Implementation of an Isolated Word Recognition System

by

Elias Mehretab Hagos

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

SYSTEMS ENGINEERING

January, 1985

Order Number 1355757

Implementation of an isolated word recognition system

Hagos, Elias Mehretab, M.S.

King Fahd University of Petroleum and Minerals (Saudi Arabia), 1985

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

University of Petroleum and Minerals

IMPLEMENTATION OF AN ISOLATED WORD RECOGNITION SYSTEM

BY

ELIAS MEHRETAB HAGOS

A Thesis Presented to the
FACULTY OF THE COLLEGE OF GRADUATE STUDIES

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

IN

SYSTEMS ENGINEERING

January 1985

Printed By: UPM Printing Press

UNIVERSITY OF PETROLEUM & MINERALS

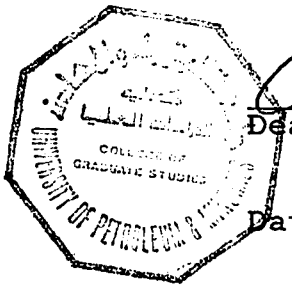
Dhahran, Saudi Arabia

This thesis, written by

ELIAS MEHRETAB HAGOS

under the direction of his Thesis Committe, and approved by all its members, has been presented and accepted by the Dean, College of Graduate Studies, in partial fulfillment of the requirments for the degree in

MASTER OF SCIENCE IN SYSTEMS ENGINEERING



Abdullah Al-Zaki
Dean, College of Graduate Studies

Date Jan. 14, 1985

Department Chairman

THESIS COMMITTEE

Md. Shahjir Ahmed
Chairman

Member

Member

ACKNOWLEDGMENTS

Acknowldegment is due to the University of Petroleum and Minerals for giving me the opportunity to complete my studies part-time while working with the University DPC full-time. I extend sincere and deep appreciation to Dr. Mohammed S. Ahmed who introduced me to the area of pattern recognition and served as my major advisor. I also wish to thank Dr. Ala H. Al-Rabeh, Chairman of the Systems Engineering Department and Dr. Shokri Z. Selim both of whom served in my thesis committee.

I would also like to extend my deepest gratitude to the DPC for its superb facilities. The thesis report was written using the SCRIPT document processor with the 6670 printer. The speech data transfer from the PC to the Mainframe was accomplished using the DPC IBM PC-XT that is connected to the IBM 3033 via IRMA board.

ABSTRACT

A speaker-independent Arabic digits recognition system is implemented which uses template matching of input utterances with a stored set of multiple templates for each digit. The system is based on the LPC parameters for features, the log likelihood ratio for a distance function between frames, the procedure of dynamic time warping for time-normalization between test and reference utterances and the K-NN rule for decision criterion.

Four utterances for each digit from each speaker were collected to form a data-base of 80 replications for every word. The reference templates were obtained from a statistical clustering analysis of this data-base.

Two implementations were considered; in one approach the LPC features are extracted at a fixed rate while in the other approach they are extracted at a variable rate by merging similar neighbouring fixed size frames. Both implementations were tested against a three category test data-base, (a) utterances used to train the system, (b) utterances not used to train the system but by speakers used in the training phase and (c) utterances from speakers not used in the training phase.

خلاصـــــــــة

لقد تم انجاز نظام لتمييز أو للتعرف على نطق الارقام العربيـــة والتى تحت رقم العشرة لمتكلم ما ، حيث يستخدم هذا النظام عدة مساطـــر عينات من نطق تلك الأرقام كمدخلات للنظام مماثلة لمساطر عينات متعـدد ة ومختزنة لكل رقم من الارقام التى تحت العشرة • ويعتمد هذا النظـــــام على معاملات التنبوء الخطى لنطق الكلام ، وعلى سجل نسبة الاحتمـــــال لدالة المسافة بين اطارات مناطـــق مختلفة تمثل نطق الكلام ، وعلـــــى الاجراء الديناميكى لعملية المقارنة بين المناطق المعدة للفحص والمناطق المعيارية التى تستخدم كمرجع فى المقارنة وكذلك على قانون ك ـ أقـــرب نقطة مجاورة لمعيار القرار •

لقد تم جمع نطق كل رقم تحت العشرة أربعة مرات من كل متكلـــــم لتشكيل قاعدة بيانات مكونة من ٨٠ تكريرة لكل كلمة ، مساطر العينـــات المستخدمة كمرجع للمقارنة تم الحصول عليهم من تحليل مجاميع احصائية من قاعدة البيانات التى تم تشكيلها •

لقد تم اعتبار وسيلتين فى هذا النظام ، احدى هاتين الوسيلتـــين هى معاملات التنبوء الخطى لأقســـام النطق الكلام تم استخلاصهم بمعدل ثابـــت بينما فى الوسيلة الثانية تم استخلاص أقسام نطق الكلام بمعدل متغـــير وذلك بدمج أقسام نطق الكلام المتماثلة والتى من نفس الحجم •

هذا وقد تم فحص كلتا الوسيلتين مقابل ثلاثة مجموعات من قواعـــد فحص البيانات (أ) بعض نطق هذه الكلمات لتدريب النظام •(ب) بعـــــض نطق هذه الارقام لم يستخدم لتدريب النظام ولكن من متكلمين استخدمـــوا فى مرحلة التدريب ، (ج) بعض نطق هذه الارقام من متكلمين لم يستخدمـوا فى مرحلة التدريب •

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

BACKGROUND AND LITERATURE REVIEW

1.1 INTRODUCTION.

Man's interest in voice communication with machines has a long history [1]. However, it is only during the last thirty years that theoretical advances in the fields of digital signal processing, phonetics, acoustics, electronics, artificial intelligence and pattern recognition and technological break-throughs in the area of digital computers have brought about impressive developments in all aspects of speech processing by machines. The increasing complexity and variety of the machines that man has to interface with is a driving force for theoretical and technological develepments in the area of man-machine communication. The most advanced of this interaction between man and machine is, of course, manifested in the various means of interfacing with the digital computer. In all the traditional means of controlling machines, VDU, keyboard, paper outputs etc., the human operator is required to adopt to the machine's modes of input or "language". These intermediate agents of communication hamper the ease and versatility of controlling machines as any deviation from their inflexible rules produces errors. The introduction of voice input/output capability simplifies

communication since it is the machine that will be required to adopt to the language of the human.

Some of the advantages of using voice communcations [2] include the effective use of human communicative abilities, its use in unusual circumstances and the possiblity it affords to be used concurrently with other activites. Thus it is likely to have wide applications [3] and bring about revolutionary changes if and when machines are able to reliably use voice input/output.

The general field of man machine communication by voice can be logically divided into three subareas, voice response systems, speaker recognition and speech recognition

Voice response designates those applications in which the machine, in response to a specific set of conditions, communicates pertinent information to the human by voice [4]. The activating set of condidtions might be a request for information by the operator in inquiry systems or unusual events that the machine is designed to monitor and issue warning messages by voice whenever they occur. An example of this applications is the speaking typewriter that utters the keys depressed as an aid to the blind. The distinguishing aspect of this area is that the voice communication is in only one direction, from the machine to the human.

The area of speaker recognition, on the other hand, is an application of pattern recognition principles that uses the speech utterances from a speaker to either verify his claim of his identity (speaker verification) by comparing his

utterance to stored features from the claimed speaker [5], or identifies the speaker (speaker identification) by comparing his voice with a set of stored voice features of a class of speakers [6]. In both cases, the direction of the voice communication is from the human to the machine.

Speech recognition, the most complex form of man-machine interaction, deals with the situation in which the machine having received a voice input responds in a manner consistent with the intention of message in the input. It has received wide research attention [7-10]. The next section will deal with various aspects of this area.

1.2 Speech Recognition

1.2.1 Introduction.

Unlike the two other voice communication modes in which the problem is well defined, the area of speech recognition allows for a wide range of options to determine the nature and complexity of the application of interest. It is possible to define a number of complexitiy factors such as form of the speech, isolated words or continuous speech; the scope of the recognizer, speaker independent or tailored to a speaker; the type of the speaker, cooperative, indifferent or uncooperative; the size and nature of the vocabulary, small, large, general or task specific; speaking environment, noisy or clean, etc. [11,12]. In fact, various subfields have sprouted depending on the above mentioned factors. In spite of the diversity of options, the field

can be viewed as logically being divided into either isolated word speech recognition (IWSR) or continuous speech recognition (CSR) [13].

The complexity of the problem has hampered rapid progress. The diversity of speech signals, the enormous variability among different speakers and even for the same speaker at different times introduce great difficulties in reliably extracting invariant parameters. The speech acoustic signals upon which ultimately all recognition must depend contain a great deal of redundant information. One way of dealing with this complexity is to limit the scope of the problem to consider only isolated words of limited vocabulary. The consideration of only isolated words simplifies the identification of the word boundary problems which is one of the most difficult tasks to automate.

1.2.2 Statistcal Pattern Recognition Scheme

The area of isolated word recogniton can be viewed as a particular case of the classical pattern recognition problem [14]. The classical pattern classification scheme has two phases, the development or training phase and the implementation or use phase.

In the former phase patterns whose class is known are used to train the system. In its simplest form, training consists of measuring, processing, clustering and storing relevant features of this known patterns. Since in their original form, the patterns are usually very high dimensional, they are processed so that only features that

are sufficient to discriminate among the different pattern classes are selected. This procedure both reduces the dimensionality of the patterns and improves the recognition performance by removing information that is not useful to differentiate between classes. In an attempt to remove the effect of variability among the different occurences of the same class, statistical techniques are used to cluster the samples in the training phase. The operation of clustering identifies the representative patterns for a class and which are stored as reference templates for use in the classification stage. It also is a dimension reduction procedure in that only a fraction of the samples in the training stage are retained for reference purposes.

In the second phase, when the system is in operation, as a pattern of unkown class is received, the system compares it with all the templates stored in the training phase and based on scores obtained from the comparison classifies it as belonging to class to which it is closest. Thus, this phase involves the definition of similarity measures to compare different patterns and decision rules for classifying unknown pattern based on evaluation of the similarity measures.

1.2.3 ISOLATED WORD RECOGNITION

To put the isolated word recognition system in the above framework, therefore requires three things; the preprocessing and selection of features from the speech acoustic signals, the determination of what distance

measures to use in order to compare two words and the establishment of decision strategy to be used.

The analog acoustic signal is first low pass filtered so that high frequency components, not significant for recognition, are removed. The resulting signal is then digitized at a rate greater than twice the cut-off frequency of the low pass filtering. Automatic algorithms are needed to detect the end points of the words since the manual procedure is both undesirable and unreliable [15-16]. Usually, a preemphasis filtering is applied at this stage for spectral smoothing purposes. The acoustic signal is now in a form from which the features can be selected.

Among the various features that have been used are time domain features such as zero crossing rates, energy measurements and bandpass filter coeficients. The more suitable features are, however, the frequency domain measurements which include spectral coeficients obtained by DFT, cepstral coefficients and the linear prediction coeficients (LPC).

LPC modelling, where for short intervals comparable to the pitch period the speech signal can be assumed to be the output of a linear time-invariant system, has been extensively used because of its conceptual and computational simplicity. The digitized acoustic signal is blocked into sequences of units, called frames, short enough to justify the assumption of stationarity and long enough to enable the estimation of the autocorrelation coefficients which are used in the evaluation of the LPC. In this technique the

speech signal is estimated as a linear combination of the last p speech signals where values of p between 8 and 14 are frequently mentioned in the litrature. The LPC are obtained by minimizing the sum of the sqared error between the estimated and true values [17-19].

The LPC have an intimate relationship with linear models of the speech production mechanism. They represent the coefficients of the all pole model of the vocal tract both for voiced and fricative sounds over pitch period time durations and thus provide smoothed estimate of the spectral content of the speech signal.

It is possible to perform word template matching only if there is a quantitative measure of similarity between features of frames of utterances derived by acoustic analysis. Once such a measure is established, the features derived from the unknown word can be compared with all the stored templates for classification. Some of the distance measures used are spectral distances, Euclidean distances between the time domain signals, covariance weighted distances and the LPC log likelihood measure (Itakura measure) [20,21].

But due to variations in speed among speakers and errors in automatic detection of end points of utterances, the different utterances to be compared are usually of different time duration. Thus, when two utterances are compared, it is necessary to time align them so that the frames being compared with one another can be assumed to correspond to similar events. The distance between the two utterances is

then the sum of distances between frames for some optimal time warping of the utterances.

A frequently used scheme for time-warping two utterances is the dynamic programming time warping algorithm. In this scheme, the test and reference patterns are aligned by performing a dynamic programming optimization to select the mapping that yields the least distance between the utterances [20,22-24].

Once a suitable distance measure for the pattern space has been defined, clustering algorithms are then used in the training stage to identify the representative protoype patterns to be stored for later use. This helps in making the recognition process independent of the speakers as the clustering procedure is designed to remove speaker characteristics from the reference templates [25-27].

The last aspect of pattern recognition invloves the decision strategy. After comparing the unknown pattern with the reference templates some form of the nearest neighbor(NN) rule is ususlly employed to classify the unknown pattern [14]. Typically, the input pattern is identified as belonging to the class to which it is nearest. A rejection threshold may also be used in this stage to reject any pattern as not belonging to any of the stored template classes if the distance between the input pattern and its nearest template exceeds a pre-established threshold value for that class.

The pattern matching isolated word recognition scheme discussed above has been used by a number of researchers.

Itakura [20] describes a system that is based of the LPC computed from windowed samples taken every 15ms. He also introduced the log likelihood distance measure between two speech frames and employed dynamic time warping procedure in his single speaker 200 word recognition system. Sambur and Rabiner [28] used a combination of features, zero crossing rates, energy and LPC residual error pole frequency. A speaker independent recognition system that uses the LPC is described by Rabiner et.al, [29]. Other implementations are given in [30,31]. An excellent tutorial of the field can be found in [32]. References [12,13,33] provide a rich review of different implementations and comparison of attained performances.

An approach that differs from the template matching procedure described in the preceding paragraphs is the feature-based recognition approach that is based on human perception principles. The idea is to use various features, distributed across time and frequency, about various phonetic events and the manner they covary [34].

1.2.4 Continuous Speech Recognition

Continuous Speech Recognition (CSR), the ultimate goal of speech recognition research, deals with the situation in which the input to the machine comes in the manner humans normally use speech for communication. One of its distinguishing charactersitics is that the objective is to understand the speech and not merely recognize it as is the case with isolated word recognition. When a person

recognizes speech, he uses much more information than is contained in the acoustic waveform. Such higher level knowledge of the syntax and semantics of the language and the subject matter of the speech contributes a great deal to the recognizability of continuous speech. Thus, for machines to perform as well as humans, they must be provided with such capabilities. These requirements create considerable difficulties for various reasons.

Although the human speech organs can produce an unlimited variety of sounds, the number of distinct sounds, called phonemes, for a given language ranges between 30-50. Thus, it seems at first sight, that knowledge about the acoustic characteristics of the phonemes of a language is sufficient to enable machine recogintion. The problem with continuous speech is that the acoustic characteristcs of a word exhibit great variations depending both on the sentence it is in and the words that precede and follow it. This contextual dependence, known as coarticulation, creates an enormous variety of occurences for a given sound in continous speech which makes it impossible to perform a word-by-word analysis in CSR.

A second problem with automating CSR is that the sophisticated linguistic rules a human uses to recognize speech is not well understood and thus can not be programmed on machines. Furthermore, the human uses a broad knowldege base about the subject matter of the discourse to deduce meanings that are implied by the speech but that are not explicitly expressed. A third problem that is particularly

severe for continous speech is the variability, in speed and pronounciation, that exists among different speakers.

One way to deal with these complexities is to restrict the problem. Some of the ways to limit the capability of the machine is to restrict the size of vocabulary, restrict the syntax of the language to an artificially constructed grammar, to limit the scope of discourse to a specific application and to limit the number of speakers. The research studies so far conducted incorparate some form of these restrctions.

A concentrated research effort in the CSR problem was conducted in the 1970's sponsered by the Advanced Research Projects Agency (ARPA) of the Department of Defense of the United States. The initial objective of that project was to develop systems that accept from cooperative but varied speakers continuous speech of an artificial syntax language consisting of a vocabulary of 1000 words and understand the speech in real-time with an upper error rate of 10%. A detailed description of the nature and objectives of that project is in [35].

By 1976, that research effort lead to the development of various systems that fulfilled the intial objectives. The four successfully devloped systems are the HWIM (Hear What I mean) of the Bolt, Beranek & Newman Inc. [36], the Systems Development Corporation (SDC) Speech Understanding Project [37], and Harpy and Hearsay-II both developed at the Carnegie Mellon University [38,39]. A tutorial overview of the entire ARPA Speech Understanding Project is given in [40].

Another significant research into CSR is the on-going efforts by IBM researchers to develop a speech understanding system [41-43]. Their approach, which uses ideas from communication theory, is different from other attempts in that it is based on statistical modeling of all speech processes and recognition is attempted by using the input signal to hypothesise possible words maximizing the likelihood that a sentence construced from the hypothesised words best matches the input sentence.

One of the major results to emerge from these studies is a better understanding on how to use syntatic and semantic knowlege to assist in the recognition process [44]. They have also demonstrated that speech understanding is possible when the form of the continuous speech is restricted.

Although the complexities of CSR have slowed down the volume of research, the Japanese fifth-generation computer project envisions the construction of a voice-activated typewriter with a vocabulary of 10,000 to operate in a speaker-independent mode. Such an ambitious task has to tackle various problems, however. Some of these problems are, the introduction of better techniques than the template matching by dynamic programming approach for large vocabularies, the search for more discriminatory features to guarantee speaker-independence, the construction of cheap VLSI chips with powerful computational capabilities to handle the enormous amount of computational load and the use of knowldge aquired in building expert systems using artificial intelligence to enable the systems graceful error recovery to resolve perceptual confusion [45].

1.3 Thesis Proposal

The vocabulary of most of the isolated word recognition systems that have been developed is taken from English words. Though, there is no reason to believe that the particular structure of a word recognizer may be language dependent, the difference in phonetic structure may affect its performance. Thus, in this thesis, a speaker-independent isolated word recognition system for the Arabic digits zero to nine using the LPC as features and the Itakura measure as the distance function is implemented. The templates will be derived from a database of the words using statistical clustering techniques. Two segmenting appraoches, frames of fixed size and frames of variable size, will be considered.

CHAPTER II
SPEECH PRODUCTION AND LINEAR PREDICTION

2.1 INTRODUCTION

In a rather simplified view of the human speech communication process four stages can be identified; the encoding of a message into an acoustic wave by the vocal tract by a speaker, the propagation of this wave through a channel usually air, the reception and processing of the acoustic signals by the ear and auditory nerves of the listner and finally the decoding and eventual perception of the message. The last two stages belong to the receiver which consists of the human functions of hearing and understanding the message. In designing an automatic speech recognition system, ASRS, there are four approaches that may be adopted corresponding to each of the four stages of the human communication process, the generation, propagation, reception and perception of messages.

The simplest approach is to argue that since the message is contained in the acoustic signals, all that is needed is the analysis of these signals using general signal processing techniques to extract relevant feature vectors and then use mathematical and statistical techniques to template match the input speech signals against stored prototype patterns to asses their similarity and classify

them accordingly. The distinguishing characteristic of this approach is that no prior information about either the way the signals are produced or the manner in which they are received and perceived by humans is used. They are treated just like any other signals and it is the apporach usually employed in general recognition applications. Its basic simplicity makes it a compelling method to consider. However, the lack of information beyond the acoustic signals is a severe design limitation.

In the second approach, the speech production modeling approach, a parametric form is used in which the signal is represented as the output of an assumed model of speech production. The measured acoustic signals are analysed in order to derive the parameters of a model of the process that is assumed to have produced the signal. The message, in this case, is not represented by the time-amplituted values as in the first approach but rather by the derived parameters of the speech production model. Since the parameters of the speech production process change at a lower rate than the acoustic waveform itself, the use of such prior knowledge to encode the speech message results in a substantial representational economy. In other words a small number of parameters of the speech production model is sufficient to represent the message and the recognition process employs template matching principles on the model parameters to identify the message content. Although, there is a loss of information due to this representation, such a

loss is not particularly severe for speech recognition applications.

The third approach attempts to duplicate the manner in which the human ear and auditory nerves process the acoustic signals so as to base the recognition process on features similar to those that are derived by the human ear when it receives acoustic signals.

In the last approach the ASRS is based on the manner in which the human perceives the messages carried by the acoustic signals. Systems based on this approach combine information derived from the acoustic signals with higher level knowledge about the linguistic and semantic structure of speech. It requires the understanding of the manner in which the human brain uses the signals as received by the ear to decode the messages they contain.

The four design approaches outlined are not mutually exclusive, of course. In practice, a combination of some or even all of them may be employed. The first two approaches differ only in the manner of representation of the speech signal. They have found wide applications in speech recognition. The last two approaches are more difficult to implement because the processes they are based on, hearing and understanding, are not as well understood as the processes of the first two approaches.

The LPC based ASRS that was implemented essentially uses the speech production modeling technique. This chapter will, thus, be devoted to an exposition of both the

mechanism of speech production and the rather successful method of deriving the model parameters by the autocorrelation method of linear prediction.

2.2 Speech Production Mechanism

The human speech production mechanism is a complex process that involves organs extending from the lungs, trachea, vocal cords, larynx, the oral and nasal cavities to the lips and nose. A schematic diagram of the anotomical parts involved in this process is given in figure 1.
The source of energy for the entire process is the pressure due to the air exhaled from the lungs. During the production of sound, air, exhaled from the lungs, flows through the glottis, the pharynx and the oral or nasal cavities to be radiated through the lips or the nostrils. The resonating tube consisting of the larynx, the pharynx and the oral and the nasal cavities is called the **VOCAL TRACT**. The velum, or soft palate, the organ which hangs from the roof of the back of the mouth, is a connecting door that couples the nasal and oral cavities during the production of nasal sounds.

The pressure variations in the air flow causes the vocal cords, a loose membrane at the top of the trachea, to vibrate which in turn modulates the airflow creating a quasi-periodic pulse-like pressure wave that excites the vocal tract during the creation of the voiced sounds such as /a/ in the word apple. The frequency at which the vocal

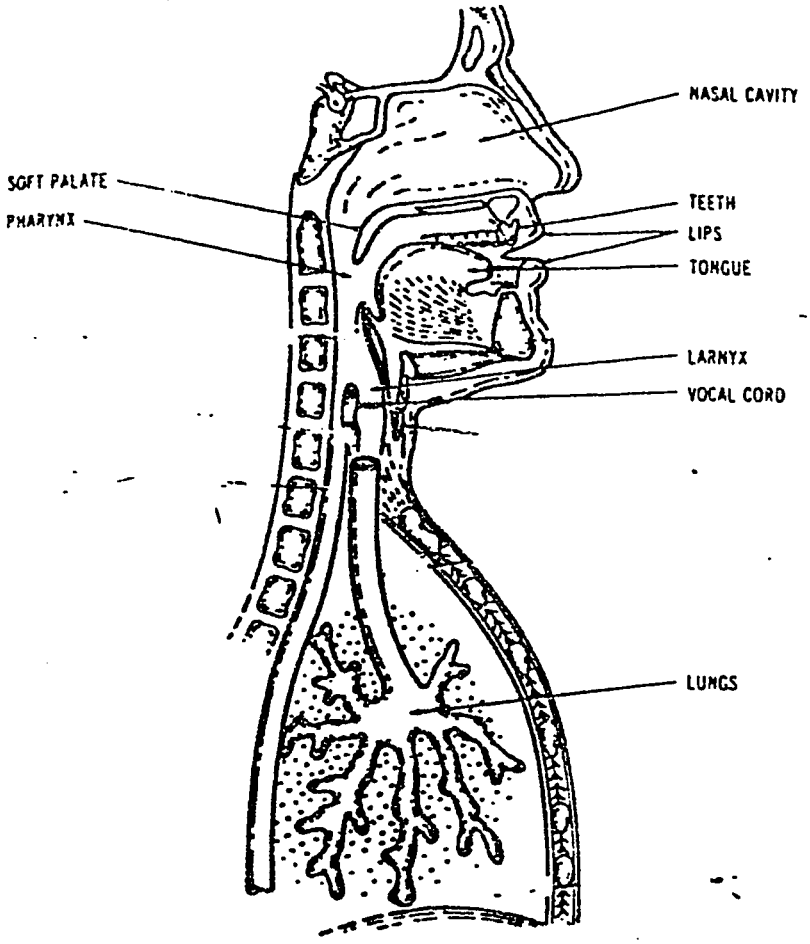


Figure 1: The human speech production organs.

---

cords vibrate corresponds to the fundamental frequency and is related to the acoustic property identified as pitch of the sound. The vocal tract, being a resonant cavity, modifies the frequency content of the quasi-periodic wave, amplifying some frequencies and attenuating others. The vocal tract resonant frequencies are known as the FORMANT FREQUENCIES, or in short formants. The velum, tongue, the lips and jaws, the speech articulators, are used to change

the shape of the vocal tract which causes corresponding changes in the formant frequencies and thus the voiced sounds are produced.

If the vocal tract is constricted somewhere in the oral cavity, the relatively smooth air flow is disturbed to produce turbulent flow. This creates a random noise-like source that excites the vocal tract to produce sounds known as fricatives. Depending on where the constriction occurs, different sounds are produced. If the vocal cords are relaxed, unvoiced fricatives such as /f/, and /s/ result while voiced fricatives such as /sh/, /j/ are created when it vibrates.

A third type of sound are the plosives which are produced when the air flow is completely restricted by closing the vocal tract so that pressure is built behind the closure and then the closure is suddenly opened to release the trapped air. Various sounds of this kind are produced depending on where the closure in the vocal tract occurs. For the sound /b/, for example, the closure is at the lips while for the sound /d/ it is behind the teeth. Unvoiced plosives such as /t/ are produced when the vocal cords are held fixed during the opening of the closure. On the other hand, the voiced plosives such as /d/ are produced if the vocal cords are simultaneously vibrating with the opening of the closure.

When the velum is lowered to couple the nasal cavity to the oral cavity, nasal sounds and nasalized vowels are

created. In the production of nasal sounds, the oral cavity is closed and the sound is radiated through the nose. If the closure is at the lips the sound /m/ is created and when it is behind the teeth the resulting sound is /n/. The closed oral cavity is still a resonating tube. The resonant frequencies are, however, frequencies at which the closed oral cavity traps energy from the sound wave and the main effect of this energy entrapment is manifested as anti-resonances in the radiated acoustic wave.

It is clear from the preceding presentation that there are broadly two types of sounds, voiced and unvoiced, depending on wether the vocal cords oscillate or are held still during the production of the sound. This difference is refered to as arising due to the manner of articulating the sound. Another source for variety in the sounds that can be produced is the place where the vocal tract shapes are modified and this is refered to as the differences due to the place of articulation.

The sound wave is further modified as it is released from the lips and nostrils. Thus, the sound received at a distance from the mouth also includes the effects of this radiation at the lips and the nostrils.

The discussion has so far focussed only on the production of a single sound during which the vocal tract shape can be assumed to be fixed. Speech is, however, the concatenation of different sounds whose broad properties are well understood. These different units of sounds are called

PHONEMES and depending on the language their number may range between 30-40. A word is usually composed of more than one phoneme and the transition from one to another phoneme is achieved by changing the shape of the vocal tract. The vocal tract is, thus, a time-varying structure.

To exactly model the generation of sound would involve too much detail. Instead, a simplified model is used that captures the essential elements in the mechanism. From the preceding discussion, the entire speech process can be conceptualized to consist of four components; an excitation source, the effects at or below the glottis, the frequency shaping effect of the time-varying vocal tract and the radiation effect at the lips and nostrils. A block diagram representation of this mechanism is shown in figure 2.

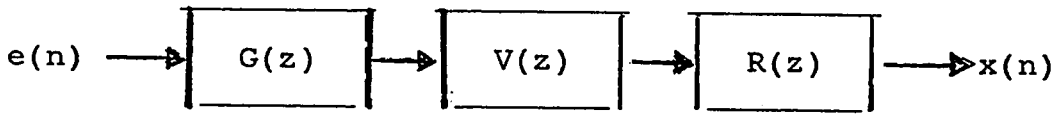


Figure 2: Block diagram representation of the speech production system

A linearized representation of figure 2 in the z-domain is expressed as

$$X(z) = E(z)G(z)V(z)R(z) \qquad [2.1]$$

where E(z) is the z-transform of the excitation source which is impulse-like for voiced sounds and noise-like for fricatives, G(z) represents the glottal and sub-glottal effects, V(z) is the resonance filtering effect of the vocal tract and R(z) is the radiation factor. The vocal tract filter, the glottal and radiation effects can be lumped together to give

$$X(z) = E(z)H(z) \qquad [2.2a]$$

where

$$H(z) = G(z)V(z)R(z) \qquad [2.2b]$$

In the preceding model the glottal shaping model, G(z), is approximated by a two-pole transfer function {11}of the form

$$G(z) = G1/(1-z_1z^{-1})(1-z_2z^{-1}) \qquad [2.3]$$

where both poles are real and inside the unit circle with the maginitude of one of the poles close to one.

The radiation effect at the lips accounts for the relation that exists between the volume velocity of the air flow at the lips and the sound pressure of the radiated acoustic wave. At low frequencies the pressure can be approximated by the first derivative of the volume velocity which leads to a reasonable approximation as a first difference in the discrete case. Thus it can be modeled as

$$R(z) = R_0(1-z^{-1}) \qquad [2.4]$$

Finally, the vocal tract transfer function is modeled as an all-pole filter of complex conjugate poles to acount for its resonant characteristics. This model ignores the presence of zeroes in nasal sounds. A typical complex pair of poles in the s-domain is given by

$$s_k, s_k^* = -\sigma_k \pm j\omega_k \qquad 1 \le k \le F \qquad [2.5]$$

where F is the number of formants covering the frequency range of interest. By changing variables $z_k = e^{-s_k T}$, where T is the sampling period, the discrete form poles are given by

$$z_k, z_k^* = e^{(-\sigma_k \pm j\omega_k)T} \qquad [2.6a]$$

$$= e^{-\sigma_k T} \; e^{\pm j\omega_k T} \qquad [2.6b]$$

$$= e^{-\sigma_k T} \; e^{\pm j\omega_k T} \qquad 1 \le k \le F \qquad [2.6c]$$

Thus, the transfer function of one pole pair is given by

$$V_k(z) = C_k \; / \; (1 - 2|z_k|\cos(w_k T)z^{-1} + |z_k|^2 z^{-2}) \qquad [2.7]$$

where

$$|z_k| = e^{-\sigma_k T} \qquad [2.8]$$

is the maginitude of the pole. The over all all-pole transfer function of F formants is

$$V(z) = \prod_{k=1}^{F} V_k(z) \qquad [2.9a]$$

$$= \prod_{k=1}^{F} \{C_k/(1 - 2|z_k|\cos(w_k T)z^{-1} + |z_k|^2 z^{-2})\} \qquad [2.9b]$$

During the production of speech which is composed of varying sounds, the vocal tract is continually changing. But due to the inertia of the speech articulators that cause this variation, it can be assumed that the vocal tract is in a steady position over short periods of time. And thus, H(z), the lumped transfer function is modeled as a linear time-varying system whose parameters for short periods of time may be considered to be constant.

This rather simplistic model of the speech production process has proven very succesful in both speech analysis and synthesis. In fact speech synthesisors such as the TI Speak and Spell, where the synthesised speech is of acceptable quality have been constructed using such model, a fact which reinforces the belief in the power of the model to capture the essential properties of speech production.

In the remainder of this chapter, we discuss the powerfull method of linear prediction, which enables us to estimate the vocal tract transfer function parameters from the digitized speech samples.

2.3 Linear Prediction of Speech

Given a speech signal sequence x_n, assumed to be the result of a stationary vocal tract, in the linear prediction analysis the sample x_n is modeled as a linear combination of its past values and an input sequence u_n, i.e.

$$x_n = - \sum_{k=1}^{p} a_k x_{n-k} + G u_n \qquad [2.10]$$

where G is some gain factor and where p, the order of the predictor filter, is the number of past samples used in predicting the n'th sample. The minus sign is used to simplify the notation later on, but otherwise is not significant. By taking the z-transform of both sides of (2.10), we get

$$X(z) = U(z) \left[G / \left(1 + \sum_{k=1}^{p} a_k z^{-k}\right) \right] \qquad [2.11a]$$

$$= U(z) \left[G/A(z) \right] \qquad [2.11b]$$

$$= U(z) H(z) \qquad [2.11c]$$

where U(z) and X(z) are the the z-transforms of the sequences u_n and x_n respectively. A(z), known as the inverse filter, is given by

$$A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k} \qquad [2.12a]$$

$$= \sum_{k=1}^{p} a_k z^{-k} \qquad \text{with } a_0=1 \qquad [2.12b]$$

Comparison of the assumed linearized model of the speech production system of (2.2) and the linear prediction model in (2.11) shows that the combined effects of the glottal wave, the radiation at the lips and the vocal tract transfer function is being modeled by G/A(z). A block diagram illustration of the linear prediction model in the time domain is given in figure 3.

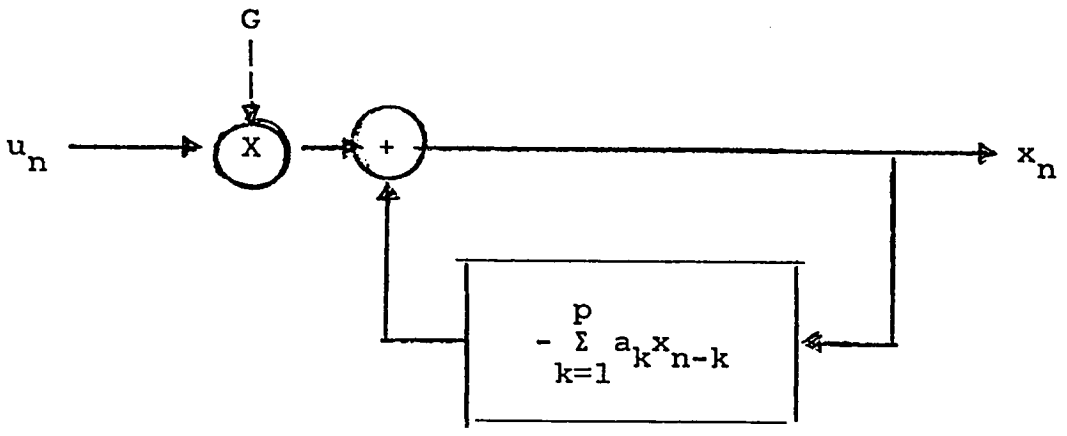


Figure 3: Speech production model in the time domain.

The discussion up to now has dealt with the model of speech synthesis. In attempting to derive a parametric representation of a given speech sequence x_n, the appropriate approach is to perform an inverse analysis on the acoustic signal x_n to estimate in some optimal sense a model of the vocal tract that produced the speech signal. This is essentially a reversal of the speech production process in which a model of the vocal tract is derived from the measured speech wave. The only available data at this inverse analysis stage is the measured signal sequence x_n. Thus, because the input sequence u_n and gain factor G are unknown at this inverse analysis stage, the signal x_n can only be predicted from a linear combination of its p past values as

$$x'_n = - \sum_{k=1}^{p} a_k x_{n-k} \qquad [2.13]$$

where x'_n is the predicted value. Upon taking the z-transform of both sides we obtain

$$X'(z) = P(z)X(z) \qquad [2.14]$$

P(z), the linear prediction filter, is given by

$$P(z) = - \sum_{k=1}^{p} a_k z^{-k} \qquad [2.15]$$

The difference between the actual value x_n and the linearly predicted value x_n is the prediction error sequence given by.

$$e_n = x_n - x'_n \qquad [2.16]$$

which upon substitution for x'_n from equation (2.13) becomes

$$e_n = x_n + \sum_{k=1}^{p} a_k x_{n-k} \qquad [2.17]$$

In the z-domain, the prediction error is given by

$$E(z) = X(z) - X'(z) \qquad [2.18a]$$

$$= [1 - P(z)]X(z) \qquad [2.18b]$$

$$= [1 + \sum_{k=1}^{p} a_k z^{-k}]X(z) \qquad [2.18c]$$

$$= A(z)X(z) \qquad [2.18d]$$

The inverse fitering action of A(z), which is in essence separating the signal spectrum X(z) into an excitation spectrum E(z) and a linear time-invariant model 1/A(z), is illustrated in figure 4.

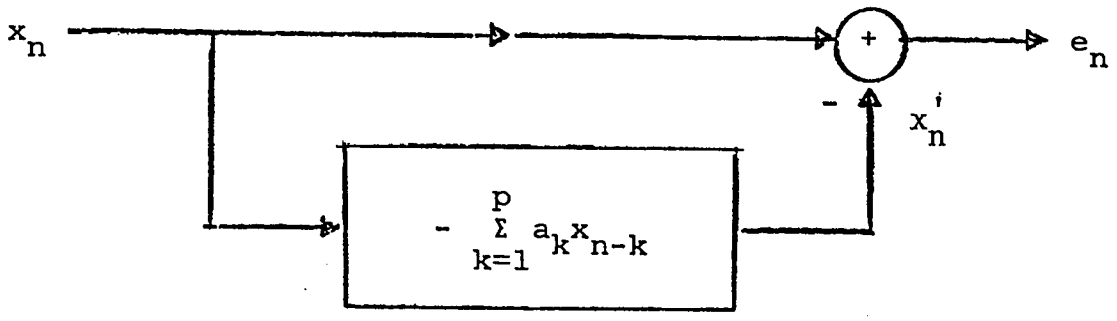


Figure 4: Inverse filtering in the linear prediction analysis.

The problem of linear prediction is reduced to the selection of the coefficients $[a_1, a_2, \ldots, a_p]$, the linear pedicition coefficients (LPC) and the gain factor G by minimizing some measure of the signal prediction error as given by (2.17) over a time interval during which the speech signal can be assumed to be stationary.

There are various functions of the error sequence of the form

$$E = \sum_n |e_n|^m \qquad m>0 \qquad [2.19]$$

that can be minimized. The function usually used because of its mathematical tractability and some other desirable

properties to be presented later is, however, the case where m is set to 2, which leads to the method of linear least squares prediction. Thus, the function minimized is given by

$$E = \sum_n e_n^2 \qquad [2.20]$$

where the range of the summation is left unspecified for the moment as it will not affect the following discussion.

By substituting for e_n from (2.17) the sum of squared errors can be written as,

$$E = \sum_n e_n^2 \qquad [2.21a]$$

$$= \sum_n [x_n + \sum_{k=1}^{p} a_k x_{n-k}]^2 \qquad [2.21b]$$

To obtain the optimal set of parameters, $[a_1, a_2, \ldots, a_p]$, the total sum of squared errors, E, is minimized with respect to each coefficient by setting the partial derivatives of E with respect to a_i, $1 \le i \le p$ to zero. The minimization yields the set of p equations in the p unknowns

$$\partial E/\partial a_i = 0, \qquad 1 \le i \le p \qquad [2.22]$$

Applying (2.22) to (2.21) we obtain

$$\partial E/\partial a_i = 0 \qquad 1 \le i \le p \qquad [2.23a]$$

$$= \sum_n 2\, e_n\, \partial e_n/\partial a_i \qquad [2.23b]$$

$$= 2 \sum_n e_n x_{n-i} \qquad [2.23c]$$

$$= 2 \sum_n [x_n + \sum_{k=1}^{p} x_{n-k}] x_{n-i} \qquad [2.23d]$$

which after some manipulation reduces to

$$\sum_{k=1}^{p} a_k \sum_n x_{n-k}x_{n-i} + \sum_n x_n x_{n-i} = 0 \quad 1 \le i \le p \qquad [2.23e]$$

The set of equations (2.23) are various forms of the **NORMAL EQUATIONS** so called because the optimal error sequence is seen to be from (2.23c) normal to the last p signal sequences $x_{n-1}, \ldots, x_{n-p}$.

Depending on the range of the summation in (2.23), two methods of solving for the prediction coefficients a_i's emerge known in the literature as the autocorrelation and covariance methods [19]. In the covariance method, the prediction error is minimzed over a finite set of points. The discussion, however, will be limited to the autocorrelation method only in which the range of the summation is from $-\infty$ to $+\infty$.

Thus, the summed values in (2.23) constitute the autocorrelation coefficients of the sequence x_n, i.e.

$$r_i = \sum_{n=-\infty}^{\infty} x_n x_{n-i} \qquad [2.24a]$$

$$r_{|i-k|} = \sum_{n=-\infty}^{\infty} x_{n-k} x_{n-i} \qquad [2.24b]$$

The optimality condition (2.23e) can be now put as

$$\sum_{k=1}^{p} a_k \, r_{|k-i|} = - r_i \qquad 1 \le i \le p \qquad [2.25]$$

The autocorrelation coefficients as defined by (2.24) are symmetric, i.e.

$$r_i = r_{-i} \qquad 1 \le i \le p \qquad [2.26]$$

The minimum total squared error of the model of order p denoted by E_p is obtained then as

$$E_p = \sum_{n=-\infty}^{\infty} e_n^2 \qquad [2.27a]$$

$$= \sum_{n=-\infty}^{\infty} e_n \left[x_n + \sum_{k=1}^{p} a_k \, x_{n-k} \right] \qquad [2.27b]$$

which upon using the orthogonality conditions becomes

$$= \sum_{n=-\infty}^{\infty} e_n \, x_n \qquad [2.27c]$$

$$= \sum_{n=-\infty}^{\infty} \left[x_n + \sum_{k=1}^{p} a_k \, x_{n-k} \right] x_n \qquad [2.27d]$$

$$= \sum_{n=-\infty}^{\infty} x_n x_n + \sum_{k=1}^{p} a_k \sum_{n=-\infty}^{\infty} x_{n-k} x_n \qquad [2.27e]$$

$$= r_0 + \sum_{k=1}^{p} a_k r_k \qquad [2.27f]$$

It must be noted that the use of the term autocorrelation in this context differs from its usual sense where the mean of the signal is removed prior to the evaluation of the summation. A matrix form of the normal equations (2.25) is

$$\begin{bmatrix} r_0 & r_1 & r_2 & \cdots & r_{p-1} \\ r_1 & r_0 & r_1 & \cdots & r_{p-2} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ r_{p-1} & r_{p-2} & r_{p-3} & \cdots & r_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_p \end{bmatrix} = - \begin{bmatrix} r_1 \\ r_2 \\ \cdot \\ \cdot \\ \cdot \\ r_p \end{bmatrix} \qquad [2.28a]$$

which in short is expressed as

$$RA = R \qquad [2.28b]$$

where $A^t=[a_1,a_2,\ldots,a_p]$ is the vector of prediction coefficients, $R^t=[r_1,r_2,\ldots,r_p]$ and R is the autocorrelation matrix.

The gain factor G is obtained by imposing the condition that the energy of the input signal spectrum to be equal to energy of the model spectrum {18-19} and will be derived later on.

The simultaneous equations given in (2.28) can be solved by inverting the matrix $\mathbf{R}$. We note, however, that the matrix $\mathbf{R}$ has the Toeplitz structure (the elements along any diagonal are equal). This structure allows for a very efficient algorithm due to Levinson and Durbin [46,47], in which the LPC's of an inverse filter of order p are recursively obtained in p steps. A derivation of this algorithm is given here both for its own merit and for the insight it affords into the nature of autocorrelation method of linear prediction.

2.3.1 Derivation of the Levinson-Durbin Algorithm

Starting from the trivial solution to the filter of order zero, the algorithm proceeds recursively to solve for the m'th order least mean square error model from the solution to the (m-1)'th order filter. It will be shown that the factor that leads the recursive solution is the even property of the autocorrelation coefficients, i.e. $r_k=r_{-k}$ for k=1,2,..,p.

Assume that the (m-1)'th order filter LPC vector has been obtained as $(A^{m-1})^t=[1,a_1^{m-1},a_2^{m-1},\ldots,a_{m-1}^{m-1}]$, where the superscript indicates the order of the model. These solution set satisifes the equation (2.28).

$$\begin{bmatrix} r_0 & r_1 & r_2 & \cdots & r_{m-2} \\ r_1 & r_0 & r_1 & \cdots & r_{m-3} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ r_{m-2} & r_{m-3} & r_{m-4} & \cdots & r_0 \end{bmatrix} \begin{bmatrix} a_1^{m-1} \\ a_2^{m-1} \\ \cdot \\ \cdot \\ \cdot \\ a_{m-1}^{m-1} \end{bmatrix} = - \begin{bmatrix} r_1 \\ r_2 \\ \cdot \\ \cdot \\ \cdot \\ r_{m-1} \end{bmatrix}$$

[2.29a]

or in short

$$R^{m-1}A^{m-1} = R^{m-1} \qquad [2.29b]$$

where R^{m-1} is the matrix of autocorrelation coefficients and $(R^{m-1})^t=[r_1,r_2,\ldots,r_{m-1}]$ is the right hand side column vector. By reversing the set of equations in (2.29) we obtain the equivalent set of equations

$$\begin{bmatrix} r_0 & r_1 & r_2 & \cdots & r_{m-2} \\ r_1 & r_0 & r_1 & \cdots & r_{m-3} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ r_{m-2} & r_{m-3} & r_{m-4} & \cdots & r_0 \end{bmatrix} \begin{bmatrix} a_{m-1}^{m-1} \\ a_{m-2}^{m-1} \\ \cdot \\ \cdot \\ \cdot \\ a_1^{m-1} \end{bmatrix} = - \begin{bmatrix} r_{m-1} \\ r_{m-2} \\ \cdot \\ \cdot \\ \cdot \\ r_1 \end{bmatrix}$$

[2.30]

To proceed to the m'th order model, let the solution be given by $(A^m)^t=[1,a_1^m,a_2^m,\ldots,a_m^m]$, where the superscript again indicates the order of the model. These parameters also satisfy the m normal equations (2.28)

$$\left[\begin{array}{ccccc|c} r_0 & r_1 & r_2 & \cdots & r_{m-2} & r_{m-1} \\ r_1 & r_0 & r_1 & \cdots & r_{m-3} & r_{m-2} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ r_{m-2} & r_{m-3} & r_{m-4} & \cdots & r_0 & r_1 \\ \hline r_{m-1} & r_{m-2} & r_{m-3} & \cdots & r_1 & r_0 \end{array}\right] \left[\begin{array}{c} a_1^m \\ a_2^m \\ \vdots \\ a_{m-1}^m \\ \hline a_m^m \end{array}\right] = - \left[\begin{array}{c} r_1 \\ r_2 \\ \vdots \\ r_{m-1} \\ \hline r_m \end{array}\right]$$

[2.31a]

which takes the matrix form of

$$R^m A^m = R^m \qquad [2.31b]$$

By partitioning the matrix equation in (2.31a) as indicated gives

$$\begin{bmatrix} r_0 & r_1 & \cdots & r_{m-2} \\ r_1 & r_0 & \cdots & r_{m-3} \\ \vdots & \vdots & \cdots & \vdots \\ r_{m-2} & r_{m-3} & \cdots & r_0 \end{bmatrix} \begin{bmatrix} a_1^m \\ a_2^m \\ \vdots \\ a_{m-1}^m \end{bmatrix} + (a_m^m) \begin{bmatrix} r_{m-1} \\ r_{m-2} \\ \vdots \\ r_1 \end{bmatrix} = - \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_{m-1} \end{bmatrix}$$

[2.32a]

and

$$\sum_{k=1}^{m-1} a_k^m \, r_{m-k} + a_m^m \, r_0 = -r_m \qquad [2.32b]$$

By pre-multiplying the matrix equation in (2.32a) by $(R^{m-1})^{-1}$, the inverse of R^{m-1}, we get

$$\begin{bmatrix} a_1^m \\ a_2^m \\ . \\ . \\ . \\ a_{m-1}^m \end{bmatrix} + a_m^m (R^{m-1})^{-1} \begin{bmatrix} r_{m-1} \\ r_{m-2} \\ . \\ . \\ . \\ r_1 \end{bmatrix} = - (R^{m-1})^{-1} \begin{bmatrix} r_1 \\ r_2 \\ . \\ . \\ . \\ r_{m-1} \end{bmatrix}$$

[2.33]

Substituting for the second term from (2.30) and for the third term from (2.29a), the equation reduces to

$$\begin{bmatrix} a_1^m \\ a_2^m \\ . \\ . \\ . \\ a_{m-1}^m \end{bmatrix} = a_m^m \begin{bmatrix} a_{m-1}^{m-1} \\ a_{m-2}^{m-1} \\ . \\ . \\ . \\ a_1^{m-1} \end{bmatrix} + \begin{bmatrix} a_1^{m-1} \\ a_2^{m-1} \\ . \\ . \\ . \\ a_{m-1}^{m-1} \end{bmatrix}$$

[2.34]

which can be put as the set of equations

$$a_0^m = 1 \qquad [2.35a]$$

$$a_j^m = a_j^{m-1} + K_m \, a_{m-j}^{m-1} \qquad 1 \le j \le m-1 \qquad [2.35b]$$

$$a_m^m = K_m \qquad [2.35c]$$

where K_m is known as the PARTIAL CORRELATION COEFFICIENTS.

We notice from (2.35) that if the partial correlation coefficients are available, then the recursion from step m-1 to m is complete. A derivation of the partial correlation coefficients that is based on the idea of forward and backward predictions is given in Appendix A. The complete recursive solution is given then as

$$a_0^0 = 1 \qquad [2.36a]$$

$$E_0 = r_0 \qquad [2.36b]$$

The solution for values for m>0 are obtained by using the recursion

$$a_0^m = 1 \qquad [2.36c]$$

$$K_m = - \left[r_m + \sum_{k=1}^{m-1} a_k^{m-1} \, r_{m-k} \right] / E_{m-1} \qquad [2.36d]$$

$$a_j^m = a_j^{m-1} + K_m \, a_{m-j}^{m-1} \qquad 1 \le j \le m-1 \qquad [2.36e]$$

$$a_m^m = K_m \qquad [2.36f]$$

$$E_m = (1-K_m^2) \, E_{m-1} \qquad [2.36g]$$

where the intermediate value E_m is the total residual at step m. Thus, the solution for the p'th order least square error filter is obtained as

$$a_j = a_j^p \qquad 0 \le j \le p \qquad [2.37]$$

The discussion has so far ignored the summation range for the autocorrelation coefficients. Since only a finite set of samples is available, the infinite summation in the evaluation of the autocorrelation coefficients cannot be performed. Instead, the signal is windowed so that it becomes zero outside the range of available data, [0,N-1].

$$y_n = \begin{cases} w_n\, x_n & 0 \le n \le N-1 \\ 0 & \text{otherwise} \end{cases}$$

where w_n is a window function to be discussed in Chapter III. The autocorrelation coefficients are then estimated from the windowed signal y_n as

$$r_k = \sum_{n=0}^{N-|k|} y_n y_{n+|k|} \qquad 0 \le k \le p \qquad [2.38]$$

The discussion that led to the recursive solution is, however, unaffected when this estimated autocorrelation coefficients are used as they still retain their symmetry.

It is clear from (2.36) that the Durbin algorithm solves the m'th order inverse filter at each step and that the total residual E_m is obtained as a by-product. Furthermore, the partial correlation coefficients, or the PARCOR as they are commonly known, are obtained as intermediate values.

They have been shown [18] to be related to the reflection coefficients of discrete acoustic tube models of the vocal tract. Specifically, when the vocal tract is modeled as a concatenation of p tubes, shown in figure 5, each of cross-sectional area A_j, j=1,2,...,p, then the partial correlation coefficients are given by

$$K_j = (A_j - A_{j+1})/(A_j + A_{j+1}) \qquad [2.39]$$

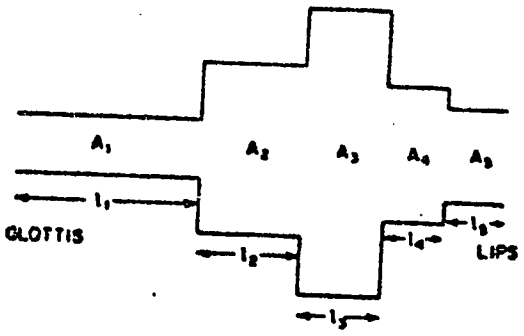


Figure 5: Acoustic tube models of the vocal tract

In addition, being correlation coefficients, they satisfy the property

$$|K_j| \leq 1 \qquad 1 \leq j \leq p \qquad [2.40]$$

which has made them very efficient parameters for speech vocoding and transmission.

The concept of the forward and backward prediction methods is not a mathematical artifice soley introduced to facilitate the solution algorithm. Having postulated that the signal x_n can be predicted as a linear combination of the p past samples, the PARCOR method first obtains the partial correlation between x_n and x_{n-1} which is the amount of information in x_n that can be accounted for by x_{n-1}. In order to derive the partial correlation between x_n and x_{n-2} the effect of x_{n-1} on both x_n and x_{n-2} is first eliminated. Application of this idea at the m'th stage provides the correlated quantity between x_n and x_{n-m}. When this correlated portion is subtracted at each stage, the residual signal is used to evaluate the partial correlation for the next stage. The LPC can then be derived recursively from the partial correlation coefficients as presented previously.

2.4 SPECTRAL INTERPRETATION OF THE METHOD OF LINEAR PREDICTION.

It was shown that the inverse filtering action produces a error sequence e_n when the signal spectrum x_n is passed through the filter A(z) whose coefficients are the LPC.

Although both the formulation and solution algorithm were presented in the time domain, the formulation in the frequency domain provides superior insight into the manner the signal spectrum is approximated by the model spectrum.

From the inverse filtering action we have

$$E(z) = X(z)\ A(z) \qquad [2.41a]$$

or

$$X(z) = E(z)/A(z) \qquad [2.41b]$$

where A(z) is the inverse filter. Conversely, if the error sequence e_n excites the filter 1/A(z) then the resulting output is the signal x_n. The method of linear prediction provides a smoothed all-pole approximation to the signal spectrum given by

$$H(z) = G/A(z) \qquad [2.42]$$

where the coefficients of the filter A(z), the LPC, are assumed to be the coefficients of the lumped tranfer function of the speech production model. The signal and model spectra can be compared by replacing z by $e^{j\omega}$ in their respective z-tranforms. Thus the signal spectrum is given by

$$P(w) = |X(e^{j\omega})|^2 = |E(e^{j\omega})|^2 \ / \ |A(e^{j\omega})|^2 \qquad [2.43]$$

and the model spectrum as

$$P'(w) = |H(e^{jw})|^2 = G^2 \; / \; |A(e^{jw})|^2 \qquad [2.44]$$

The frequency domain least square minimization formulation can be derived from the time-domain formulation. In the time-domain the total residual that is minimized is given by

$$E = \sum_{n=-\infty}^{\infty} e_n^2$$

By using Parseval's theorem, the mean squared error in the frequency domain is given by

$$E = \frac{1}{2\pi} \int_{-\pi}^{\pi} |E(e^{jw})|^2 \, dw \qquad [2.45a]$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(e^{jw})|^2 \; |A(e^{jw})|^2 \, dw \qquad [2.45b]$$

which by substituting for $|A(e^{jw})|^2$ from (2.44) results in

$$E = \frac{G^2}{2\pi} \int_{-\pi}^{\pi} \{ \; |X(e^{jw})|^2 \; / \; |X'(e^{jw})|^2 \; \} \, dw \qquad [2.45c]$$

$$E = \frac{G^2}{2\pi} \int_{-\pi}^{\pi} [\; P(w) \; / \; P'(w) \;] \, dw \qquad [2.45d]$$

Since the integrand in (2.45) is always positive, the minimization of E ensures that the resulting model spectrum will fit the signal spectrum better at frequencies where $P(w) > P'(w)$ which is to say that the spectral estimation is better at the signal spectral peaks than the valleys. This

is a very desirable property since the signal spectral peaks for voiced sounds correspond to the formant frequencies of the vocal tract. It is also intuitive, in that resonance indicates the reinforcement of the forward and backward travelling waves within the vocal tract while the method of linear prediction by hypothesising that the signal output at time n can be predicted from the past p samples works better at those frequencies in which such a reinforcement, or resonance, is significant.

Another interesting obseravation of the filtering action of the least squares inverse filter in the frequency domain is arrived at by comparing the signal and model tranfer functions. When the signal spectrum as given by (2.43) is modeled by the all pole spectrum in (2.44), the error spectrum $|E(e^{j\omega})|^2$ is being modeled by the flat spectrum signal of power G^2. Thus, when the signal sequence x_n is passed through the inverse filter A(z), derived by the least squares method, results in a flat spectrum error sequence. In other words, the inverse filter, A(z), acts as a "whitening" filter, which implies that if the signal to be modeled, x_n, is produced as a result of exciting a linear time-invariant system by a constant spectrum signal such as an impulse or white noise, then the method of linear prediction effectively deconvolves the excitation and system spectrum. We saw in the discussion on human sound production that the excitation signal in the case of voiced sounds is a periodic pulse with high bandwidth and in cases

of fricatives it is a random turbulence. Thus in both cases, we note that the excitation signal is of a broad specturm. It is this fact along with the preceding observation of the whitening action of the inverse filter that has made the linear prediction method such an invaluable tool for analysis and synthesis in speech applications.

The preceding discussion affords a straight forward manner of obtaining the gain factor G. Since the error spectrum is modeled by the constant spectrum of power G^2, matching the signal spectral power to that of the model spectrum results in

$$G^2 = E_p \qquad [2.46]$$

where E_p is the least total sqaured error expressed as

$$E_p = r_0 + \sum_{k=1}^{p} a_k r_k$$

Figure 6 shows the signal and LPC model spectra for vowel sound /a/ of 256 points (32ms).
The LPC analysis model was of order of 15. The signal spectrum was obtained by the DFT using the FFT algorithm on the signal sequence. The model spectrum was obtained by evaluating the $|G/A(z)|^2$ along the unit circle $z=e^{j\omega}$ where the denominator was evaluated by the discrete fourier transform on the augumented LPC parameters padded with zeros to achieve the required resolution, i.e. the sequence

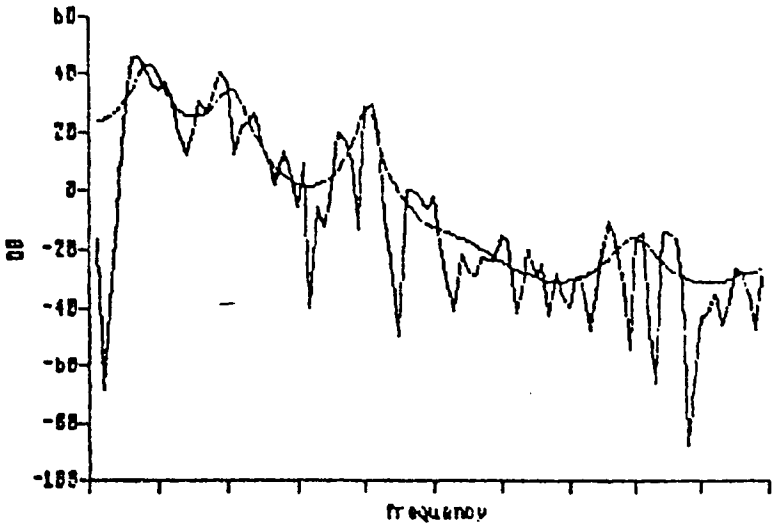


Figure 6: Input speech spectrum and the smoothed model spectrum obtained by the linear prediction method for the sound /a/.

$[1, a_1, \ldots, a_{15}, 0, \ldots, 0]$, where the total number of zeros added is 240 resulting in the total points of 256 and matchs the spectral resolution of the data sequence. The figure demonstrates the manner in which the LPC model spectrum produces a smoothed envelope of the frequency content of the signal.

It is important to note the two underlying assumptions of the linear prediction method, the assumption that the excitation is of flat spectrum and that there are no zeroes in the signal spectrum both of which are not true for speech signals. Experimental results by Atal [48] have shown

that the errors introduced by the periodicity of the excitation of voiced sounds are not severe except for high pitched sounds such as those of females and children. Although the all-pole model of the vocal tract is accurate for vowels and vowel like sounds, nonetheless, nasalized sounds and sounds where excitation source is at the interior of the vocal tract contain zeroes in their spectrum. Significant errors in the model spectrum are likely to arise when linear prediction is used. Fortunately, the experimental results with humans have shown that the ear is relatively insensitive to the zeroes which means that for speech recognition applications the modeling of the spectral poles is more important than those of the zeroes.

2.5 Analysis Conditions.

There are generally five analysis conditions {7} that have to be considered when the method of linear prediction is used, namely;

1. Sampling Frequency
2. Model Order
3. Frame Size
4. Frame Shift
5. Pre-emphasis

In this section we will discuss the selection of the model order. The other analysis conditions will be covered in chapter III in connection with the preprocessing of speech signals for the derivation of the LPC.

2.5.1 Selection of Model Order

There are two ways of establishing the appropriate model order for the linear prediction method, experimental and theoretical. It was shown in the Durbin recursive algorithm for the derivation of the LPC that the prediction coefficients and the total residual power for that step are obtained. Thus, at step m, we have a_k^m $1 \leq k \leq m$ and E_m. The residual energy is given by

$$E_m = r_0 + \sum_{k=1}^{p} a_k^m r_k$$

By dividing this equation by r_0, the zero'th order residual energy, we obtain the normalized residual energy

$$E_m' = 1 + \sum_{k=1}^{p} a_k^m r_k'$$

where the prime symbol indicate the normalization. The model order can then be selected when the decrease in the residual energy is negligible. The idea is illustrated in figure 7 which shows the normalized residual of a voiced and unvoiced sounds as a function of model order. It is seen that for model order greater than 12, the normalized residual energy is constant indicating that increasing the model order does not improve the prediction performance.

The model order can also be determined by theoretical arguments. Let the signal sequence $\{x_n\}$ be sampled at the frequency f_s. There are two acoustic waves within the vocal tract, one travelling from the glottis to the lips and the

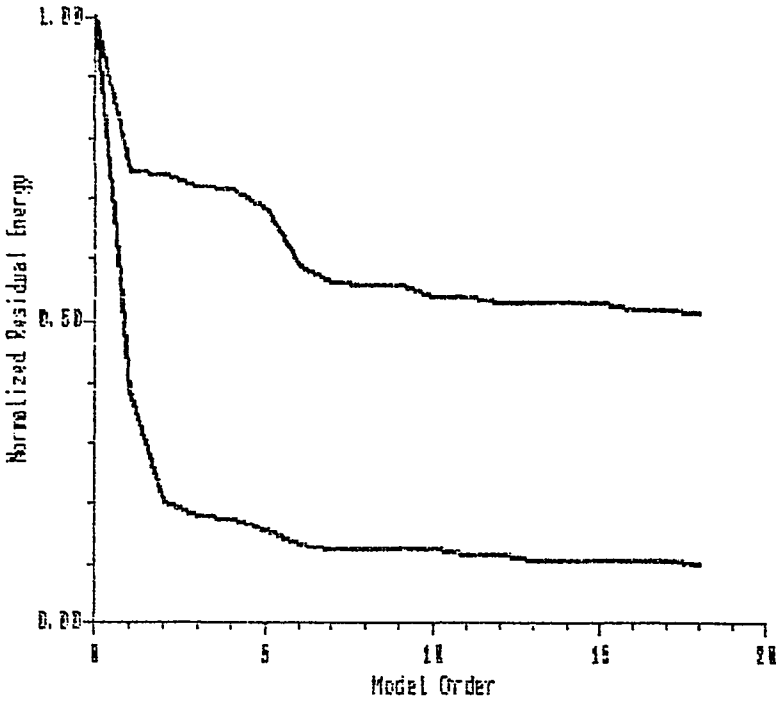


Figure 7: Normalized residual energy as a funtion of model order for a voiced and unvoiced sounds.

---

other the reflected from the lips and moving in the opposite direction. Thus if L is the length of the vocal tract, then for the sample x_n the earliest sample that can affect it is 2*L back if we ignore higher order effects and let it be given by x_{n-m}. If v is the velocity of the sound wave within the vocal tract, then the time difference between x_n and x_{n-m} is $T_d=2*L/v$. The sampling period is $T_s=1/f_s$, the number of samples equivalent to the time difference T_d, that is m, is given by

$$m = T_d/T_s = 2*L*f_s/v \qquad [2.47]$$

For example, if L=17.5 cm , v=35000 cm/s and f_s=8Khz then

m=8 which is equal to f_s given in units of Khz. These are the number of coefficients that are required to model the effects of the vocal tract. In addition two to four coefficients are required to model the glottal effects on the signal spectrum. Thus, the required model order is

$$m = f_s + g \qquad 2 \leq g \leq 4 \qquad [2.48]$$

for f_s in units of Khz.

2.6 Summary

The chapter initially discussed different approaches of designing speech recognition systems. Then, the speech production mechanism was presented and was followed by the method of linear prediction of speech. Next the efficient algorithm to solve for the LPC for the autocorrelation method was derived. The interpretation of method of linear prediction in the spectral domain was presented and finally, the selection of the model order for the method was discussed.

CHAPTER III
ISOLATED WORD RECOGNITION SYSTEMS

3.1 INTRODUCTION

A brief overview of the isolated word recognition (IWRS) scheme was presented in chapter one. In this chapter, we present in some detail, the essential elements stressing upon the fundamental pattern recognition model.

An IWRS can logically be divided into the four architectural components

1. Data Aquisition
2. Preprocessing
3. Feature Extraction
4. Utilization of extracted features

These components are implemented with a combination of software and hardware with the trend in sophisticated speech recognizers being to realize most of the tasks in hardware. Figure 8 illustrates these hardware/sortware components which also indicates the various parameters to these components and the flow of data from them.

While the scheme of figure 8 illustrates the structural view of an IWRS, a statistically implemented IWRS consists of three operational modes, mainly

1. Training (Data Collection)
2. Clustering
3. Classifying (the system is in use)

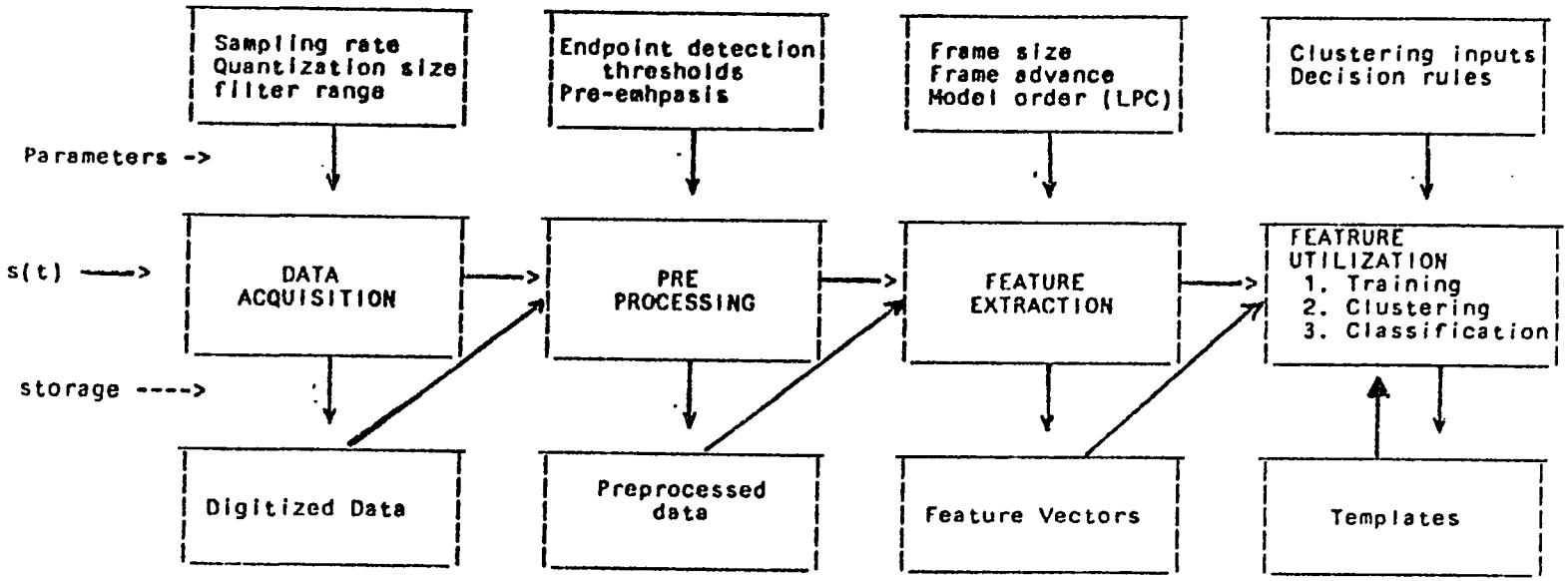


Figure 8: The architecture of an isolated word recognition system.

These operating modes differ only in the fourth structural component.

The structure of an IWRS in figure 8 allows for a wide range of design options. The decisions to be made in implementing each stage are represented as parameter inputs to the various functional blocks in the diagram. This chapter will be devoted to an illustration of these functions by dwelling on the characterisics of the Linear Prediction coefficients (LPC) based speaker-independent Arabic digits recognition system that was implemented.

3.2 Data Aquisition

The analog speech wave-form recorded or directly fed to the computer contains energies in the range 20Hz to 20KHz. In order to avoid spectral aliasing, the sampling frequency has to be at least twice the highest frequency of interest. This would require digitization at 40KHz creating an enormous data rate to deal with. In addition, the high frequency components of the acoustic wave are not significant for recognition purposes. Hence, the speech is bandlimited between 80Hz to 3.2-4KHz, the high-pass filtering around 80Hz being necessary to remove low frequency noise. The fitered signal is then sampled at 6.4-8KHz and quantized over 8-16 bits/sample. Because of difficulties in detecting the onset of speech, specially in a noisy environment, some form of manual intervention is used to start the digitization. This is not adequate for real time applications in which case the system can be

designed so that it monitors the input signal levels and starts digitization when it exceeds a threshold value. The design considerations at this stage are then the filtering frequency range, the sampling frequency f_s, the recording environment, the quantization size and threshold signal level for automatic digitization.

3.3 PREPROCESSING

The digitized signal sequence ranges from 25.2Kb/s to 80Kb/s. The speech signal contains far more information than is required to identify the utterances as is evident from the fact that a written form of a word requires far less bits to encode than its corresponding spoken form. Hence, the digital signals are preprocessed with a view of enhancing the discriminatory properties of the features to be extracted. In the context of the LPC feature extraction by the autocorrelation method, the preprocessing of interest is the end-points detection, pre-emphasis, framing and windowing.

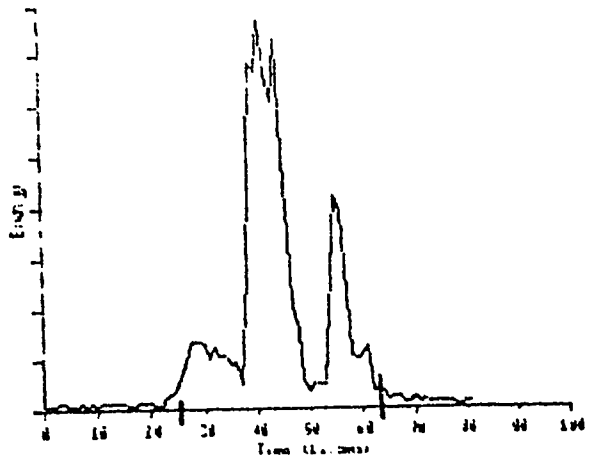
3.3.1 END-POINT DETECTION

The measured signal is usually composed of three parts; the silence part preceding the speech, the speech signal itself and the silence part that follows it. It must be noted here that by silence it is meant the background signals which depending on the recording environment could be of considerable acoustic energy. The silence parts preceding and following the speech signal have to be

discarded not only for data compression purposes but also to minimize the complications and difficulties that their presence causes to the recognition task. Hence, an automatic scheme is required to identify the beginning and ending points of the speech.

The major problem with this task is the difficulty of separating background signals from certain speech sounds specially the weak fricatives like /f/ and /s/ and the stop consonants that precede the plosives like /b/.

The approach usually implemented is to use gross signal properties such as short-time energies and zero-crossing rates to classify the signal into its three components and thus identify the end-points. Rabiner [15], has derived a robust algorithm that works in most cases. The algorithm is based on two measures of the speech signal extracted every 10ms, energy and zero-crossing rate. It obtains statistical information about the background acoustic environment from the first part of the speech signal and establishes energy and zero-crossing rate thresholds. It first uses the energy thresholds to determine an initial estimate of the end-points which is then updated using the zero-crossing rates. In figure 9, we show the application of that algorithm in the determination of the endpoints of the utterance sifr(zero).



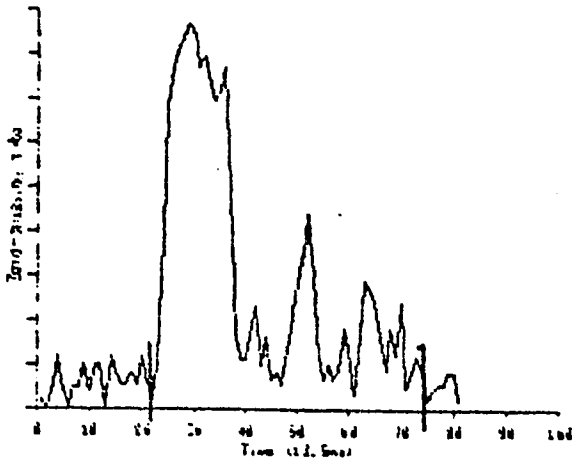


Figure 9 : Endpoints of tha Arabic word sifr(zero) derived from energy and zero-crossing rates.

3.3.2 Pre-emphasis

A pre-emphasis filter of the form $1-az^{-1}$ for values of $.9 \leq a \leq 1$ is used so that the higher frequency components are emphasized. The justification of the use of this pre-emphasis arises from the linear model of the speech production mechanism presented in chapter II where it was shown that the combined effect of the vocal tract, the glottal wave and the radiation factor to be given by

$$H(z) = G(z)V(z)R(z)$$

where G(z) is a two-pole function whose poles are of magnitute near one, V(z) is the all-pole model of the vocal tract and R(z) has one zero. The pre-emphasis has also the property of introducing a 6dB/octave function with the zero equal to one. The effect of the radiation factor zero apporximately canceles one of the glottal wave poles. The pre-emphasis is used to compensate for the other pole. A typical pr-emphasis filter is given by {32}

$$x_n = s_n - .95s_{n-1} \qquad [3.1]$$

3.3.3 Framing

The discrete signal sequence $x_0, x_1, \ldots, x_{N-1}$, where N is the number of data points available, is blocked for feature extraction purposes into F units, called frames, where each unit is of length L points. The frame length is chosen such that the stationarity of the speech signal is justified and such that the estimation of autocorrelation coefficients can

be reliably performed. Frame intervals of 10-45ms have been used in the implementation of IWRS. Consecutive frames are separated by M points where M is usually less than the frame length L so that successive frames overlap, a condition that assures smooth transitions from frame to frame. Values of M=L/3, M=L/2 and M=L are frequently used. Note that the last case corresponds to the case of no overlap between consecutive frames.

The sample points in the k'th frame are then given by

$$y_k(n) = x_{M(k-1)+n} \qquad 0 \le n \le L-1 \ , \ 1 \le k \le F \qquad [3.2]$$

where the subscript k indicates the frame number.

An obvious shortcoming of blocking the speech signal at a fixed rate is that no advantage is taken of the similarities that may exist between neighbouring frames. Since a word is usually composed of few distinct sounds and since it is very likely that acoustically homogenous contiguous frames are due the same sound source, it is logical to treat them as a single unit rather than indepenently. A small number of frames, or even one frame, may then be adequate to represent a steady sound which leads to the idea of variable frame length coding in which the speech signal is framed at fixed length to start with and then neighbouring frames are compared with one another and merged to give longer frames if they are found to be sufficiently similar. This is an implicit recognition of the fact that speech has higher level structure above that represented by the fixed size

units. By time-compressing the data by variable frame coding a large amount of redundant data is discarded. Such a compression has to be conducted with caution, however, as the redundancy may be an inherent property of the spoken word. In the English words, These and this, for instance, the elongated /ee/ sound in the first word distinguishes it from the second. In the case of isolated vocabulary recognition systems, variable frame coding can be confidently implemented if there are no occurrences of similar sounding words to warrant the above mentioned objection. The idea has been implemented for varialbe speech coding in speech transmission [49] and speech recognition [50-52]. When using the Itakura measure of log likelihood as the distance measure, a value of the likelihood ratio of 1.4 is considered a threshold for assuming speech frames to be similar [21].

3.3.4 Windowing

The framing operation introduces an implicit rectangular windowing whose effect in the time domain is to create discontinuities at the boundary points x_0 and x_{L-1}. In the autocorrelation method of linear prediction, where the signal is assumed to extend from $-\infty$ to $+\infty$, the abrupt changes at x_0 and x_{L-1} create spectral distortion. A windowing operation to taper the data at the endpoints and thus cause smooth transitions is required. The window usually employed is the Hamming window given by

$$w_n = 0.54 - 0.46 \cos\left(2\pi \frac{n}{L-1}\right) \qquad 0 \le n \le L-1 \qquad [3.3]$$

to give the windowed signal

$$y_n = \begin{cases} w_n x_n & 0 \le n \le L-1 \\ 0 & \text{otherwise} \end{cases}$$

The time-domain plot of the Hamming window and its spectral characteristics are given in figure 10.

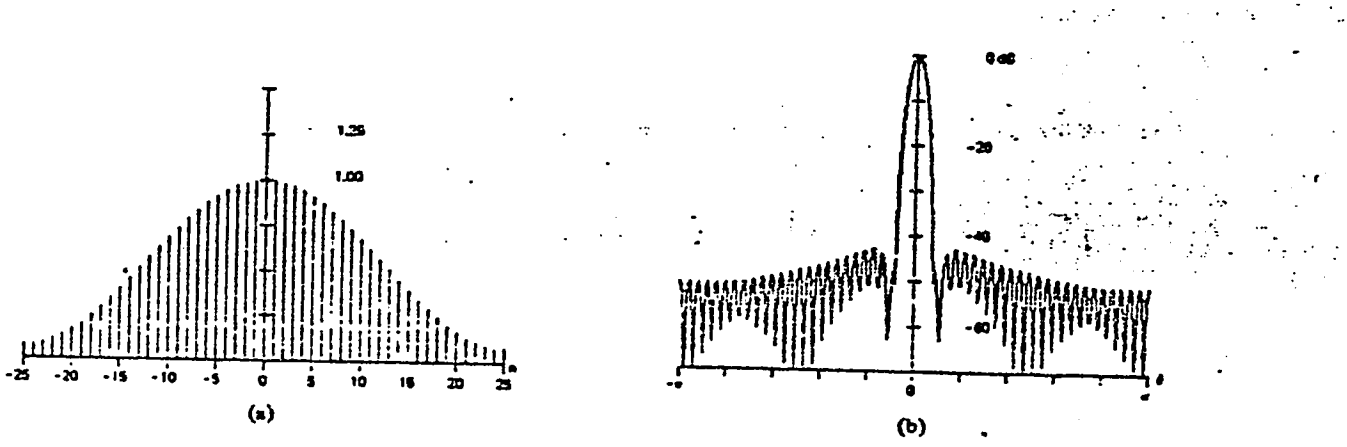


Figure 10: (a) Hamming window. (b) Log magnitude of fourier transform.

The effect of the windowing operation in the frequency domain is to convolve the signal spectrum with the window spectrum. Thus, a major consideration in window design is to compromise between the conflicting attributes of width of the main lobe and the size of the size of the side lobes. Although there is a wide range of windows that may be used [53], the Hamming window offers a good compromise of the above conditions.

3.4 Feature Extraction

In both the training and classification stages of pattern recognition, a crucial step is the selection and extraction of features. It can be confidently claimed that the recognition performance of any automatic pattern recognition system directly depends on the discriminatory properties of the features used. Moreover, In the context of speech recognition, feature selection is a further data compression stage whereby the high dimensional speech patterns are transformed into a smaller representative set of features which retain the invariant parameters in the orginal data and eliminate part of the large amount of accompanying features irrelevant to the recogniton purpose, such as those that help to identify the speaker, his emotional state, etc.

The problem of extracting a unique set of features that would enable a pattern recognition system to classify speech patterns unambiguosly has not been solved yet, however. The major difficulty with identifying features that are

invariant across speakers has to do with the great variety of the acoustic signals that pertain to a single word. In spite of these difficulties, however, researchers have come up with numerous features for speech recognition, features that are adequate if not perfect in the implementation of a limited vocabulary IWRS.

The various features that have been used in IWRS can be broadly classified as either time-domain or frequency domain features. Although the time domain features such as zero crossing rates and fundamental frequency of voiced sounds can be derived easily the more frequently used parameters are the frequency domain features because of the insight they give into the relationship between the speech signal and the manner of its articulation by the vocal organs. Consequently, short-time spectral representations such as the LPC, filter bank outputs, DFT features and cepstral coefficients have found wide use. Some of the overiding considerations in the selection of features are the goodness of representation, efficiency, minimality and ease of extraction.

The combined result of the framing and feature extraction stages is to encode the speech signal into a sequence of vector patterns in an N-dimensional feature vector space where N is the number of features used to represent every frame as $\{f_k(n)\}$, $1 \leq n \leq N$ and $1 \leq k \leq M$, where M is the number of frames.

3.4.1 Filter Bank Outputs

The digitized input sequence $\{x_n\}$ is passed through a bank of parallel band-pass filters over overlapping or distinct frequency bands. The frequency band ranges from about 100Hz to an upper cut-off frequency between 3000Hz-8000Hz. The outputs of the band-pass filters, sampled at about 40-60 samples/s are processed to yield a time-sequence of vector features $f_n(m)$, $1 \leq m \leq M$, and $1 \leq n \leq N$ where M is the number of filters and N is the number of frames. These outputs correspond to the energy of the acoustic signal over the respective frequency bands. These measurements when they are are combined with other feature sets can provide robust recogntition systems [28,54].

3.4.2 Linear Prediction Modeling.

The motivation for deriving the LPC parameters by assuming that speech is the result of exciting an all-pole digital filter by an impulse train for voiced sounds and random noise for unvoiced sounds was covered in chapter II. A particularly interesting aspect of this modeling approach is that the speech spectral estimation is reduced to the evaluation of the LPC which when using the autocorrelation method is equivalent to the problem of solving a set of simultaneous equations for which an efficient algorithm exists.

The fact that we can obtain the spectral envelope of the speech signals so efficiently makes the method of linear prediction superior to other spectrum estimation techniques such as the filter bank representation discussed previously and the short-time spectral estimation by discrete Fourier transform (DFT) using the FFT. In particular, the LPC have been very successfully used in isolated word recognition applications for the following reasons.

Firstly, The method of linear prediction deconvolves the periodic excitation from the combined spectra of the vocal tract transfer function, the radiation effects and the glottal charactersitics. In particular, when the LPC analysis is conducted after applying a pre-emphasis filter to the speech signals, the resulting all-pole spectrum is, to a very good approximation, due to the vocal tract transfer function. Thus, since during the utterance of a sound by different speakers, the vocal tract assumes approximately similar shapes, the LPC representation of the speech, by removing the harmonic structure due to the excitation, is more invariant to inter and intra-speaker variations.

A second advantage of the LPC is that the peaks of the all-pole transfer function closely correspond to the formant frequencies. When the filter order is chosen to cover the significant frequency band in the speech, a close approximation of the formants is attained which guarantees improved recognition performance.

Moreover, There exists a very efficient distance measure for comparing two speech sounds that are represented by their LPC parameters. It will be shown in the next section that this measure depends on the error residual of the linear prediction method which in the autocorrelation method is obtatined as a byproduct of the Durbin algorithm, which substantially decreases the computational load of evaluating the distance between two frames.

An appealing characteristic of the LPC's is that the entire analysis is performed in the time-domain.

3.4.3 Other Features

There are various features related to the LPC any one of which can be used for recognition purposes. These include the data autocorrelation coefficients, the PARCOR and the vocal tract area functions.

3.5 Distance Definition

The matching of utterances to establish their closeness is a fundamental aspect of pattern recognition. In fact, a major consideration in the selection of features is the requirement that the feature space allows for a definition of a distance measure. In the case of IWRS, the features are usually a time-sequence of vectors $[f_1, f_2, \ldots, f_N]$

derived over short-periods of time. Since the two utterances to be compared are generally of unequal length, there is the added requirement of defining a matching procedure for this time vectors. Such a need, in the context of speech, arises due to the variability inherent in speech utterances. Hence, similarity measure becomes a two-step procedure; firstly, the definition of a distance measure between two frames of speech and secondly having established such a measure the establishment of a procedure to match two utterances that are of different time duration.

3.5.1 Frame to Frame Distance Definition

It is possible to define distance functions both in the temporal and spectral domains. Next we present a selection of distances that have found application. In what follows, we assume the two frames to be compared are represented by the vectors $T^t = [t_1,\ldots,t_p]$ and $R^t = [r_1,\ldots,r_p]$ where the sysmbols T and R represent test and reference respectively.

3.5.1.1 Euclidean Distance

Perhaps the simplest distance function is the Euclidean distance that is given by

$$d(T,R) = \sum_{k=1}^{p} [t_k - r_k]^2 \qquad [3.4]$$

A problem with this measure is that all the components of the feature vectors are weighted equally. This introduces two disadvantages. First, the different components of the feature vectors have different variances. Thus the absolute differences between elements of feature i do not, in general, have equal significance as those between elements of feature j. The greater the variability of the components the lesser their significance should be in the distance evaluation. Therefore, a normalization by the component variance is required. Secondly, when feature elements are correlated, the result is that some of the discriminatory properties of that features contribute more to the overall distance calculation than others which tends to overemphasize their significance.

3.5.1.2 Covariance Weighted Distance

This measure, also known as the Mehalanobis distance, is a variant of the Euclidean distance that eliminates its disadvantages by weighting the distance with a covariance matrix obtained from replications of the given utterance. It has the form

$$d(T,R) = (T-R)^t C^{-1} (T-R) \qquad [3.5]$$

where the ij'th element of the covariance matrix C is given by

$$c_{ij} = E\{r_i r_j\} - E\{r_i\} E\{r_j\} \qquad 1 \le i,j \le p \qquad [3.6]$$

where the symbol E represents the expectation operation. The expected values can be approximated by averaging as

$$E\{r_i r_j\} = \frac{1}{N} \sum_{j=1}^{N} r_i \, r_j \qquad 1 \le i,j \le p \qquad [3.7a]$$

and

$$E\{r_i\} = \frac{1}{N} \sum_{j=1}^{N} r_i \qquad 1 \le i \le p \qquad [3.7b]$$

from N replications of the utterance. This distance measure eliminates the effects of any correlation between the feature components and thus all features are given equal weight in the overall distance evaluation.

3.5.1.3 Spectral Distance

Given two speech frame sequences $[x_0, x_1, \ldots, x_{N-1}]$ and $[y_0, y_1, \ldots, y_{N-1}]$, let the corresponding LPC parameters computed from them to be $A^t=[1, a_1, \ldots, a_p]$ and $B^t=[1, b_1, \ldots, b_p]$. It was seen form the derivation of the LPC that they represent the all-pole model of the resonant effect of the vocal tract and glottal structures to the volume air flow during the production of speech and that this model represents the signal spectral envelope of the signal spectrum. Thus, a conceptually simple distance between LPC A and B in the frequency domain is given by the mean square of the log spectral difference between the power spectra of the all-pole filters represented by A and B. If

$$H_a(z) = \frac{G_a}{1 + \sum_{k=1}^{p} a_k z^{-k}} \qquad [3.8)$$

and

$$H_b(z) = \frac{G_b}{1 + \sum_{k=1}^{p} b_k z^{-k}} \qquad [3.9)$$

are the all-pole models, the model spectrum is then evaluated from the transfer function by letting $z=e^{j\omega}$ where ω is the angular frequency as

$$P_a(w) = |H_a(e^{j\omega})| = G_a^2 \,/\, |A(e^{j\omega})|^2$$

$$= G_a^2 \,/\, |1 + \sum_{k=1}^{p} a_k e^{-j\omega}|^2 \qquad [3.10a]$$

and

$$P_b(w) = |H_b(e^{j\omega})| = G_b^2 \,/\, |B(e^{j\omega})|^2$$

$$= G_b^2 \,/\, |1 + \sum_{k=1}^{p} b_k e^{-j\omega}|^2 \qquad [3.10b]$$

The mean square of the log of the spectral difference is then given by

$$[d(A,B)]^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} [\log (P_a(\omega)) - \log (P_b(\omega))]^2 \, d\omega \qquad [3.11]$$

This is simply the mean square log spectral difference. Since the integrand is always positive, the distance will be zero only when the two spectra are equal.

The spectral distance as given by (3.11) is approximately calculated by replacing the integral by a summation of the difference between the the log spectra of the two frames evaluated at discrete frequencies. These discrete spectra are obtained by the DFT on the LPC's of the two frames by using the FFT where any number of zeroes can be added to the LPC's to achieve any desired resolution. It is noted here that the spectra of the signals need not be obtained through LPC analysis. The spectral distance can also be obtained by directly applying the FFT on the signal sequences or on the spectral representation obtained through filter bank approximations.

3.5.1.4 Log likelihood ratio distance

Another dissimilarity measure is the log likelihood measure orginally proposed by Itakura [20] in which two utterances are compared by computing the ratios of

the linear prediction residual energy that result when one of the utterances is filtered by the inverse filter derived from the second utterance. This measure has found wide use because of its reliablility and its computational simplicity.

Let E_a^x denote prediction residual energy (the sum of squared errors) when the sequence $\{x_n\}$ is passed through the inverse filter whose LPC is the vector $\mathbf{A}$. Consequently, given the two sequences $\{x_n\}$, $\{y_n\}$, and the LPC parameters derived from them, $\mathbf{A}^t = [1,a_1,\ldots,a_p]$, $\mathbf{B}^t = [1,b_1,\ldots,b_p]$ respectively, we define the four prediction residuals E_a^x, E_a^y, E_b^x, E_b^y. Figure 11 shows the inverse filters $\mathbf{A}^t$ and $\mathbf{B}^t$ and the four possible residuals. E_a^x and E_b^y are the minimum residuals for the inverse filters $\mathbf{A}$ and $\mathbf{B}$ respectively. i.e.

$$E_a^x < E_b^x \qquad [3.12a]$$

$$E_b^y < E_a^y \qquad [3.12b]$$

From maximum-likelihood arguments, Itakura derived what

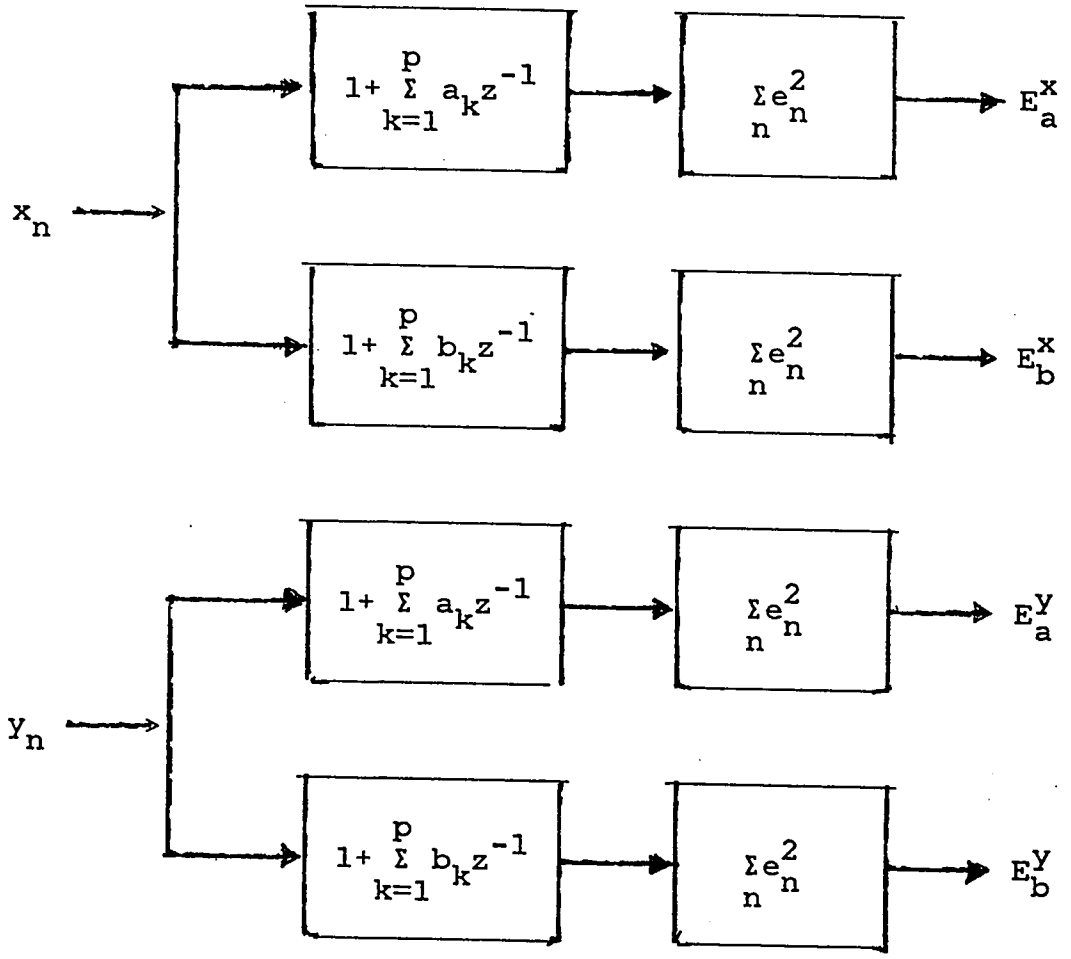


Figure 11: The four possible residuals of two data sequences

is known as the log likelihood ratio similarity measure for two speech frames with LPC parameters **A** and **B** derived by the autocorrelation method of linear prediction.

$$d(A,B) = \log [E_b^x / E_a^x] \qquad [3.13]$$

From (3.12a), it is evident that the ratio E_b^x / E_a^x is always greater than one and is equal to one only when the inverse filter **B** is identical to **A**. A similar line of reasoning leads to another form of the log likelihood ratio of the two residual energies that result when the signal $\{y_n\}$ is passed through its mean square error minimizing inverse filter **B** and **A** to give

$$d(B,A) = \log [E_a^y / E_b^y] \qquad [3.14]$$

The similarity measure defined by (3.13) and (3.14) is not symmetric i.e.

$$d(A,B) \neq d(B,A) \qquad [3.15]$$

Next, details of computational simplifications will be presented. Filtering the speech samples $\{x_n\}$ by a p-order inverse filter $A^t=[1,a_1,a_2,\ldots,a_p]$ results in the error

sequence $\{e_n\}$. The residual energy can be expressed as

$$E_a^x = \sum_n e_n^2 \qquad [3.16a]$$

$$= \sum_n \left[\sum_{i=1}^{k} a_k x_{n-k} \right]^2 \qquad [3.16b]$$

$$= \sum_n \sum_{i=0}^{p} \sum_{j=0}^{p} a_i x_{n-i} \, a_j x_{n-j} \qquad [3.16c]$$

$$= \sum_{i=0}^{p} \sum_{j=0}^{p} a_i \, a_j \sum_n x_{n-i} \, x_{n-j} \qquad [3.16d]$$

$$= \sum_{i=1}^{p} \sum_{j=1}^{p} a_i \, a_j \, r_{|i-j|} \qquad [3.16e]$$

where in the last step we have used the definition of the autocorrelation

$$r_{|i-j|} = \sum_n x_{n-i} \, x_{n-j} \qquad [3.17]$$

and the range of the summation n is left unspecified for generality. (3.16e) can now be expressed in matrix form

$$E_a^x = A^t R_x A \qquad [3.18a]$$

Similarly we have

$$E_b^x = B^t R_x B \qquad [3.18b]$$

R_x, the autocorrelation matrix of the sequence $\{x_n\}$ whose ij'th element is equal to $r_{|i-j|}$, is

$$\begin{bmatrix} r_0 & r_1 & r_2 & \cdots & r_{p-1} & r_p \\ r_1 & r_0 & r_1 & \cdots & r_{p-2} & r_{p-1} \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ r_{p-1} & r_{p-2} & r_{p-3} & \cdots & r_0 & r_1 \\ r_p & r_{p-1} & r_{p-2} & \cdots & r_1 & r_0 \end{bmatrix}$$

[3.19]

Therefore, we note that the quadratic form $A^t R_x A$ represents the energy of the residual e_n when the sequence $\{x_n\}$ whose matrix of autocorrelation coefficients is given by R_x is passed through an inverse filter with the augumented LPC parameters $A^t=[1,a_1,a_2,\ldots,a_p]$. In particular, if the LPC parameters $\{a_k\}$ are derived from the sequence $\{x_n\}$, then the resulting error sequence $\{e_n\}$ has the minimum possible energy given by $A^t R_x A$.

By inserting (3.18) in (3.13), we obatin a simplified form for the Itakura measure between two frames as

$$d(A,B) = \log [\ B^t R_x B \ / \ A^t R_x A\ ] \qquad [3.20]$$

Direct evaluation of (3.20) involves a large number of multiplications. In general, the quadratic expression $A^t RA$ requires $(p+1)^3$ multiplications where $(p+1)$ is the dimension of the matrix R. By taking advantage of the special Toeplitz structure of the autocorrelation matrix R_x, however, the amount of computation can be considerably reduced. Specifically,

$$E_a^x = A^t R_x A \qquad [3.21a]$$

$$= \sum_{i=0}^{p} a_i \sum_{j=0}^{p} r_{|i-j|} a_j \qquad [3.21b]$$

By letting the index k=i-j, we have

$$A^t R_x A = \sum_{i=0}^{p} a_i \sum_{k=i-p}^{i} r_{|k|} a_{i-k} \qquad [3.21c]$$

which can be more conveniently expressed as

$$A^t R_x A = \sum_{k=-p}^{p} r_k \sum_{i=0}^{p-|k|} a_i a_{i+|k|} \qquad [3.21d]$$

$$= r_0 \sum_{i=0}^{p} a_i a_i + 2 \sum_{k=1}^{p} r_k \sum_{i=0}^{p-|k|} a_i a_{i+|k|} \qquad [3.21e]$$

$$= r_0 f_0 + 2 \sum_{k=1}^{p} r_k f_k \qquad [3.21f]$$

where f_k is the autocorrelation of the LPC parameters $A^t=[1,a_1,a_2,\ldots,a_p]$ and is given by

$$f_k = \sum_{i=0}^{p-k} a_i a_{i+k} \qquad 0 \le k \le p \qquad [3.22]$$

It is clear from (3.21f) that the number of multiplications required in the evaluation of the quadratic expression $A^t R_x A$ has been substantially reduced to only p+1 multiplications when the autocorrelation coefficients of the LPC, f_k, are precalculated and stored.

3.5.2 Time Alignment

As was mentioned earlier, the evaluation of a distance between two utterances requires a mechanism of identifying the corresponding frames to be compared. Specifically, let the speech utterances u1, u2 are represented by the time-pattern features [A(1),A(2),...,A(N)] and [B(1),B(2),...,B(M)] respectively. The A(n)'s and B(m)'s may be, for instance, the LPC parameters for the frames of u1, n=1,2,..,N and of u2 m=1,2,..,M. But in general they could be any feature vectors that allow for a distance measure. In general N, the number of frames of u1 is not equal to M, the number of frames of u2 even though both u1 and u2 may be utterances of the same word. Such variations arise mainly due to the intrinsic variability within speakers and difficulties in end-point detection. Hence, given sets of time-patterns of features, the question of time-alignment is how to select an optimal correspondence between the frames A(n) of u1 and B(m) of u2 such that in the distance evaluation D[u1,u2], the frames being compared can be assumed to correspond to similar sound sources. Figure 12 shows two utterances of the Arabic word sifr (zero) by different speakers.

The time-graphs of the two utterances clearly demonstrate the considerable difference that can exist between utterances of the same word. In fact, at first glance, they seem to be totally unrelated. A sophisticated technique is, thus, required perform local distortions in one or the other of the utterances so that timing differences between the two

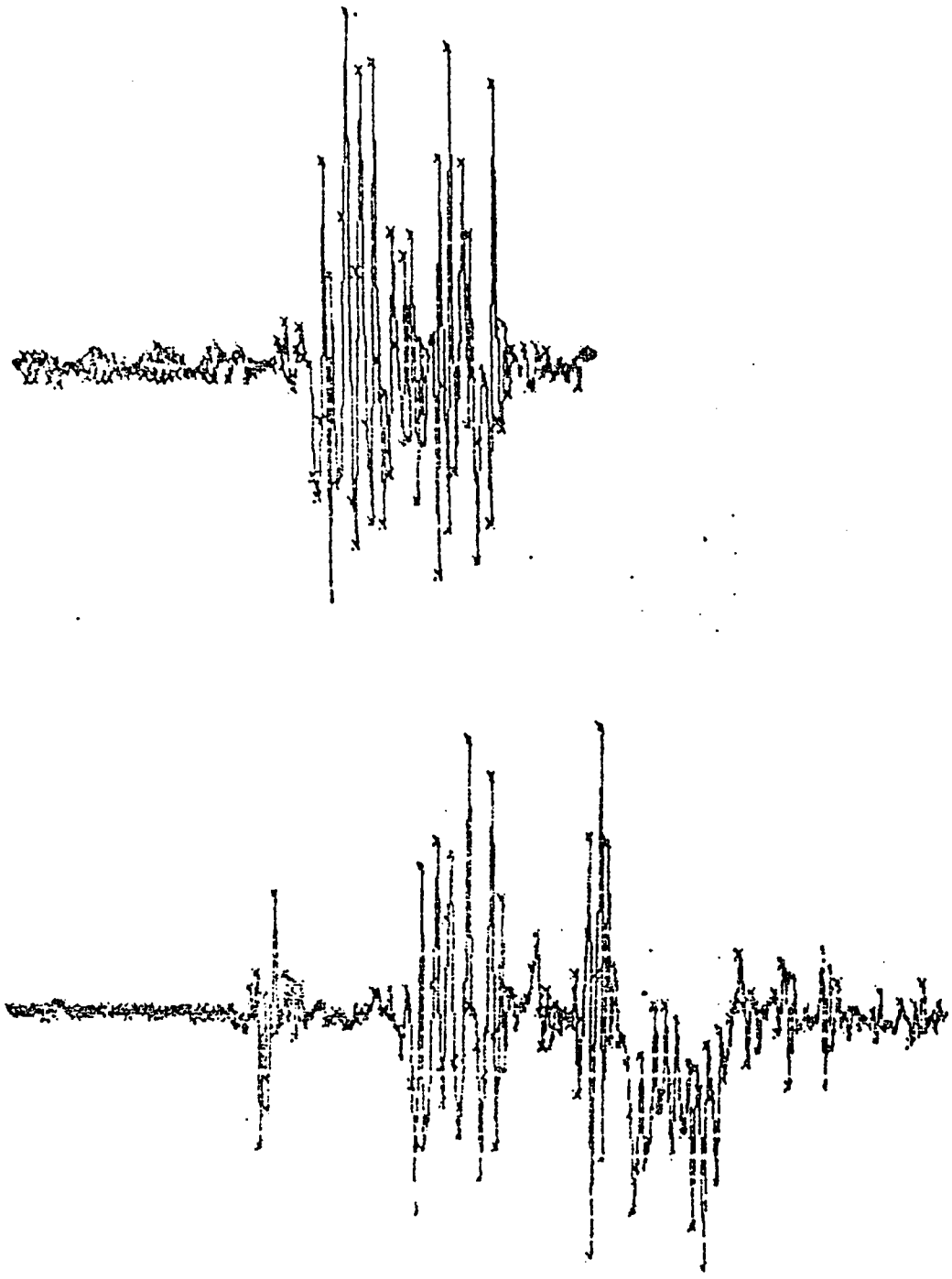


Figure 12: Two utterances of the Arabic word sifr (zero) by two speakers.

utterances can be compensated for. Figure 13 depicts the time-warping concept between two time-graphs.

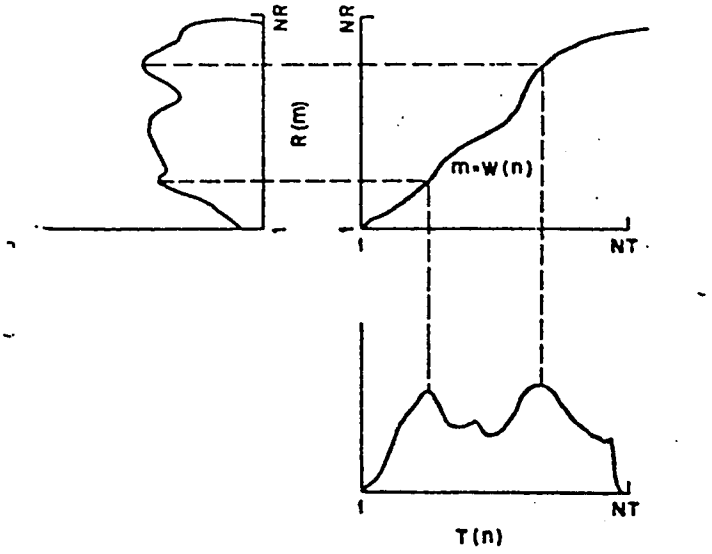


Figure 13: time-warping of two time graphs

There are various approaches to achieve the time-normalization, Here we briefly mention some of them and discuss in some detail the dynamic time-warping (DTW) method. In the following discussion the extent of utterance one and two will be assumed to be 1,..,N and 1,..,M respectively.

3.5.2.1 Linear Time Alignment

The simplest method to use is to distort one of utterance as a linear function of the other. Thus, if the time index of u2 is assumed to be generated as a linear function of that of u1, then the time-warping function can be given as

$$m = W(n) = (n-1)(M-1)/(N-1) + 1 \qquad [3.23]$$

An obvious short-coming of this solution is the unreasonable assumption that the distortions in an utterance are uniform.

3.5.2.2 Time Warping by Event-matching.

This is a modification of the previous method in which siginificant events in the two utterances such as phoneme transistions are first identified and matched. Linear time-warping is then applied to the intermediate frames. Thus, if there are k significant events for utterance one and two given as $N_1, N_2, .., N_k$ and $M_1, M_2, .., M_k$ respectively, then the event matching is put as $M_1=W(N_1), M_2=W(N_2), .., M_k=W(N_k)$. The linear time-warping in the intermediate values between event k and k-1 is then given by

$$m = W(n) = (n-N_{k-1})(M_k-M_{k-1})/(N_K-N_{k-1}) + M_{k-1}$$

$$N_{k-1} \leq n \leq N_k \qquad [3.24]$$

Figure 14 illustrates this idea. A disadvantage of this method is the complexity of identifying significant envents in speech which introduces tremendous computational load.

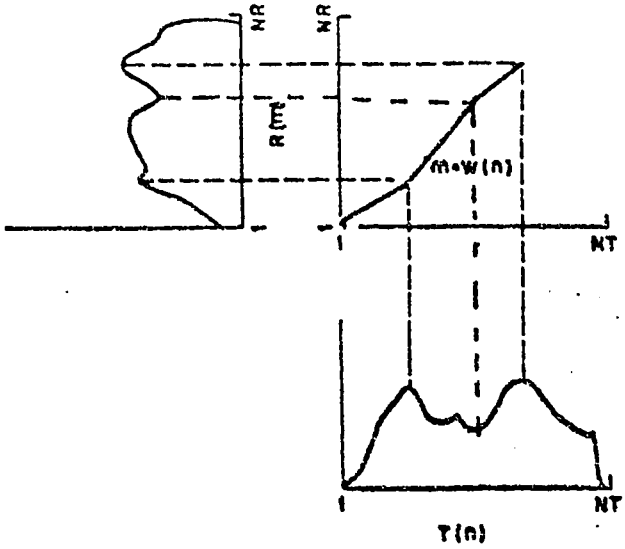


Figure 14: time-warping by event matching and linear time-warping in between.

3.5.2.3 Dynamic Time Warping

The most widely used scheme to time warp two patterns is the method that employs dynamic programming whereby an input speech signal is matched to the stored reference template by optimizing, along a coincidence path, some overall similarity measure between the two utterances. A particular correspondence path between the frames of u1 and u2 can be expressed as a set of points in the n-m plane as depicted in figure 15 and is given by

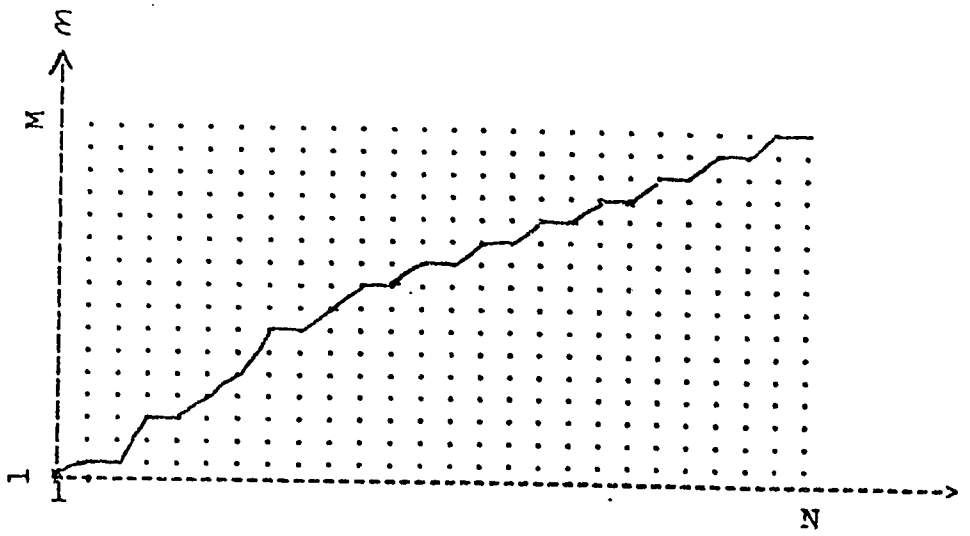


Figure 15:
A typical time-warping path.

$$f_j = \{ (n_1,m_1),\ldots,(n_k,m_k) \} \qquad [3.26]$$

The overall distance between the two utterances along this correspondence path is evaluated as

$$D[f_j] = \sum_{i=1}^{k} d[n_i,m_i]\, w_i \;/\; \sum_{i=1}^{k} w_i \qquad [3.27]$$

where d(n,m) denotes the local distance between the n'th frame of u1 and the m'th frame of u2, which might, for instance, be the log likelihood ratio given by (3.21) and w_i is a weighting function. Thus, the distance between two utterances is seen to be a functional that depends on the particular frame assignment function of (3.26). The evaluation of the optimal distance can then be formulated as

$$D[u1,u2] = \min_{j} D[f_j] \qquad [3.27]$$

where the minimization is carried over all possible path assignments in the n-m plane that are subject to various empirical and physical constraints. (3.27) is a general symmetric formulation of the time-alignment solution through optimization where both the arguments n,m in the correspondence function f_j are mapped against a common time axis. This requirement can be relaxed and a simplified assymetric formulation derived in which one of the utterances is mapped against the other in the generation of the conincidence function f_j. Specifically, if u1 is assumed to be the independent utterance, then the correspondence assignment can be realized as a mapping W(n) from the time-scale n of u1, $1 \leq n \leq N$ to that of u2, $1 \leq m \leq M$, that is m=W(n). Such a function generates a path in the n-m plane given by

$$f_j = \{ (1,W(1)),\ldots,(N,W(N)) \} \qquad [3.28]$$

where the path generation is again subject to various constraints. The recursive dynamic programming algorithm is then formulated as

$$D[n,m] = d[n,m] + \min_{j \leq m} D[n-1,j] \qquad [3.29]$$

where D[n,m] is the minimum accumulated distance to the

point (n,m) in the n-m space. The solution to (3.29) generates through n-m coordinate space the set of corresponding frames { (n,W(n) }, $1 \leq n \leq N$, such that along this path, the accumulated distance D[N,M] given by

$$D[N,M] = \sum_{n=1}^{N} d[n,W(n)] \qquad [3.30]$$

is minimal. The time warping of (3.30) is subject to three types of constraints.

3.5.2.4 Boundary Costraint

The first type of constraint comes from the need to align the endpoints. If it can be assumed that there are no errors in the endpoints, this constraint can be expressed as

$$W(1) = 1 \qquad [3.31a]$$

$$W(N) = M \qquad [3.31b]$$

Such an assumption is not realistic, however, as there is usually some uncertainty about the endpoints. A more pragmatic boundary constraint that includes such uncertainties is

$$W(1) = N1 \qquad 1 \leq N1 \leq \delta+1 \qquad [3.32a]$$

$$W(N) = N2 \qquad M-\delta \leq N2 \leq M \qquad [3.32b]$$

where δ, the uncertainty factor, can range up to five frames [27]. Note that the case when δ is equal to zero corresponds to the situation of no uncertainty in the endpoints of (3.31).

3.5.2.5 Monotonicity constraint

As a consequence of the physical nature of the speech process, the frames of the speech samples are strictly ordered in time. This physical requirement constrains the mapping W(n) to be a monotonic function, i.e.

$$W(n1) \geq W(n2), \qquad n1>n2 \qquad [3.33]$$

3.5.2.6 Slope Constraints.

The main purpose of time alignment is to overcome the time-scale variability inherent in speech utterances. The local distortions in the utterances are non-linear, that is, the rate at which sounds within the utterance change is variable. The speech time structure changes depend on the changes in the shape of the vocal tract during the utterance of the word. Thus, the dynamic programming slope constraint must account for local variations in the speech structure which might occur between utterances of the same word while at the same time disallowing those distortions that are not physically possible. Such variations can only be determined by experimenation and various constraints of this kind have been proposed and implemented [24]. A widely used form is given by

$$W(n) = W(n-1) + k, \qquad n>1$$

$$k=0,1,2 \quad \text{if } W(n-1) \neq W(n-2)$$

$$k=1,2 \quad \text{if } W(n-1) = W(n-2)$$

[3.34]

This constraint assures that the mapping W(n) does not assign the same frame m of u2 for more than two consecutive frames n of u1. The implication of this type of constraint is to limit the search space in the dynamic optimization by eliminating paths that can be assumed not to correspond to vocal tract changes. In figure 16 we show the search paths for the slope constraint (3.34) and endpoint constraints (3.32a) and (3.32b).

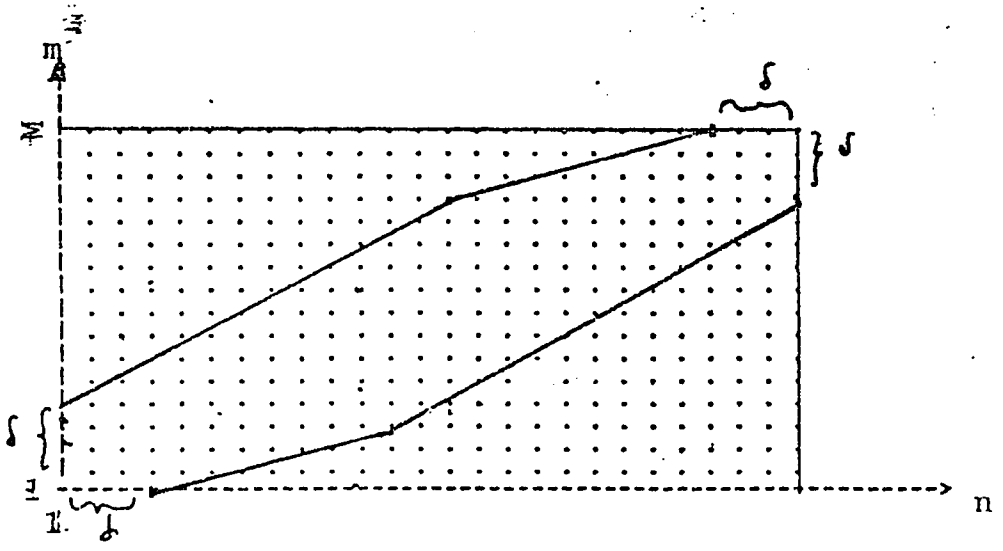


Figure 16: The dynamic programming search space.

It is evident that the distance calculation is nonsymmetric, that is, the choice of the utterance which controls the time-warping path (the utterance that is mapped to the independent axis) affects the total distance calculated, D[u1,u2]. This is due to assymetry in the slope constraint (3.34). A simple but inefficient way to obtain a symmetric distance is to evaluate both D[u1,u2] and D[u2,u1] and define the distance between the two utterances to be the average of the two. It is also possible to formulate the path constraint so that the distance calculated is symmetric. Myers, et.al [24] have experimented with various such formulations. Another approach used to minimize the effect of the assymetry is to map the test pattern to the independent axis when, in the recognition stage, it is time-warped with all the stored references. Thus, any bias produces equal effects in all the distance calculations.

The last consideration in the distance calculation is the selection of the weighting function. The usual approach for the assymetric formulation is to let $w_i=1, 1 \leq i \leq N$, which effectively results in the division of the total minimum distance D[N,M] by N, the length of the independent utterance, u1, that is controlling the time-warp. The averaged distance is then

$$D_A(N,M) = \frac{1}{N} D[N,M] \qquad [3.35]$$

The averaging normalization produces a distance that reflects the average inter-frame distance between the two utterances.

3.6 CLUSTERING

Because a single utterance cannot capture all the variations in pronounciations of the utterances, multiple templates are required to reference a word. This implies that a large number of utterances will be required to be stored in the design of speaker indpendent IWRS which introduces unbearable computational and storage load on the system. One way to circumvent this difficulty is to use features that exhibit low variability across utterances of different speakers. The search for such features has, however, met with great difficulties. To date no single set of features exist that do not vary considerably from utterance to utterance. In the absence of such features, a second alternative is to collect utterances from different speakers that encompass the range of possible occurences of a word and use statistical clustering methods.

The purpose of clustering is to choose a small set of templates that can be used to represent a large number of replications of each word in the vocabulary. It is an essential part of all pattern recognition systems that are based on the statistical behaviour of patterns. In particular, when the set of patterns of a given object can be partitioned into a number of close clusters, it is possible to represent these patterns by one pattern representative of the set. Within each cluster, the patterns have the characteristics that they are similar and are dissimilar between different clusters. The notion of sililarity is of course based on the distance measure

employed. The idea of clusters is depicted in figure 17, where a collection of patterns of two-dimensional features are shown to form two prominent clusters. The figure also shows some patterns that do not lie close to the two classes and form what are known as outliers.

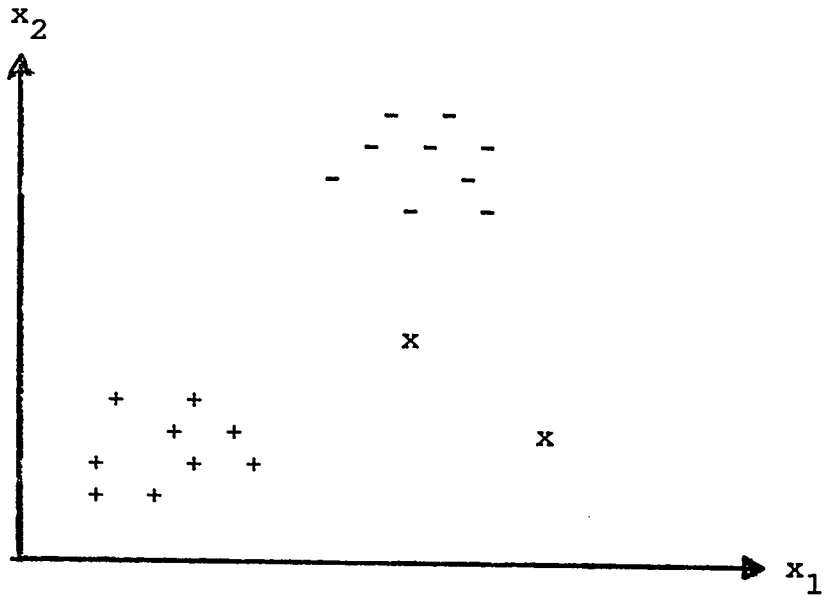


Figure 17: Two-dimensional patterns showing two clusters and outliers.

Given a set of N utterances of the same word given by

$$C = \{x_1, x_2, \ldots, x_N\} \qquad [3.36]$$

where the x_j is a time-sequence of feature vectors representing the j'th utterance, for instance the LPC, the end result of the clustering algorithm is to partition this

set into a set of M disjoint cluster sets $\{\omega_k,\ 1 \le k \le M\}$ such that

$$C = \bigcup_{k=1}^{M} \omega_k \qquad [3.37]$$

Some of the popular algorithms are the K-means and the ISODATA algorithms. The K-means algorithm requires the number of clusters M to be specified apriori while the ISODATA algorithm incorporates the possibility of merging and splitting intermediate clusters where the experimenter manually intervenes to guide the algorithm to perform the merging and splitting. Since, the number of clusters is, in general, not known apriori, the K-means algorithm is ususally not convenient to apply. The necessity of operator intervention in the ISODATA algorithm, on the other hand, makes is time consumming to employ specially in the case of a large number of patterns. In an attempt to overcome these difficualties, Rabiner and Wilpon [27] designed a number of fully automatic algorithms which they successfully used in the design of a speaker-independent IWRS. Here, a brief review of one of their algorithms, the unspervised clustering without averaging (UWA) algorithm, is given. Since the clustering is based on the similarity between the elements of the set to be clusterd, the first step in the algorithm is to obtain the matrix of distances between utterances D, where for the LPC feature vectors, D_{ij} is the time warped distance between utterance X_i and X_j. In order to have a symmetric distance matrix D_{ij} is given by

$$D_{ij} = \frac{1}{2}\,[d(x_i,x_j) + d(x_j,x_i)] \qquad 1\le i,j\le N \qquad [3.38a]$$

and

$$D_{ii} = 0 \qquad 1\le i\le N \qquad [3.38b]$$

The algorithm proceeds sequentially, at each step identifying a cluster, removing the identified cluster from further consideration and repeating the procedure until all the elements in the original set have been associated with some cluster. In order to simplifiy the discussion, we introduce the notation C^{j+1} to reperesent those patterns remaining after removing from the original set of patterns the patterns assigned to the clusters $\omega_1,\omega_2,\ldots,\omega_j$, i.e.,

$$C^{j+1} = C - \bigcup_{k=1}^{j} \omega_k \qquad [3.39]$$

Thus, C^{j+1} represents the set of elements that remain to be clustered after the first j clusters have been formed and let n_j represent their number. The clustering algorithm is given then as

1. j=0; $C^0 = C = \{x_1,x_2,\ldots,x_N\}$; $n_0=N$; $\omega_0=\phi$
2. Set j=j+1
3. Set $C^j = C^{j-1} - \omega_{j-1}$
4. Set n_j = number of elements in C^j
5. If $n_j = 0$ Stop
6. Set k=0

7. Obtain the minimax center of C^j as y_j^0 which is the pattern in the set C^j whose maximum distance to any other element in the same set is minimum.

8. Initialize the j'th cluster set as the elements in C^j whose distance from the minimax center y_j^0 is less or equal to a preset threshold radius T. Denote this set as ω_j^0.

9. Perform the following iteration KM times.

   a. $k = k+1$

   b. Obtain the minimax center of ω_j^{k-1} as y_j^k which is the pattern in the set ω_j^{k-1} whose maximum distance to any other element in the same set is minimum.

   c. Update the j'th cluster set as the elements in C^j whose distance from the minimax center y_j^k is less or equal to T and denote it by ω_j^k.

   d. If $\omega_j^k = \omega_j^{k-1}$ go to step 10 otherwise go to step 9a.

10. Store the minimax center y_j^k as a reference template of the j'th cluster set.

11. Go to step 2.

A block diagram of this algorithm is given in the Appendix B. The inputs to this algorithm are the distance matrix D, the number of replications to be clustered N, the threshold value T and the maximum iteration number KM.

In the preceding algorithm, it was implicitly assumed that the cluster centers are obtained as minimax centers for the cluster. This is a reasonable assumption to make for features such as the LPC vectors. It is also possible, in general, to average the features of the patterns within a cluster in order to determine the standard templates. Experimental results by Rabiner et al.,[27] have, however, shown the former method to be superior when the LPC feature vectors are used.

3.7 Decision Strategy

Decision strategy refers to the set of rules applied to classify an unknown utterance. Given M words in the vocabulary of the IWRS where each word is represented by ω_k, k=1,2,..,M, templates, the input utterance T is dynamically time-warped against all the stored utterances. The optimal distances are evaluated as $D[T,R_{kj}]$,k=1,2,..,M and $1 \leq j \leq \omega k$ where the index k, identifies the word k and index j identifies the j'th utterance of the k'th word.

In the simplest decision rule, the nearest neighbour(NN) rule, the the minimum of $D[T,R_{kj}]$ is found and the input

utterance is classified as coming from the word that resulted in this minimum. If the n'th utterance in the k'th word yields this minimum, then the NN-rule decides the class of the unknown input utternace as k such that

$$D[T,R_{kn}] \leq D[T,R_{ij}] \qquad 1 \leq i \leq M \text{ and } 1 \leq j \leq \omega_i \quad [3.40]$$

A more sophisticated algorithm is the K-NN rule where for each word the average distance of the K-nearest tempates to the input utterance T is evaluated and the minimum of this average for all the words determines class of T. If the distances of the reference templates of the i'th word are sorted as

$$D^*[T,R_{i1}]<D^*[T,R_{i2}]<,..,<D^*[T,R_{iK}] \quad 1 \leq i \leq M \qquad [3.41]$$

then the average distance of the K-nearest neigbours to T is given by

$$d_i = \frac{1}{K} \sum_{j=1}^{K} D^*(T,R_{ij}) \quad 1 \leq i \leq M \qquad [3.42]$$

and the input utterance is identified as the k'th word such that

$$d_k \leq d_i \qquad 1 \leq i \leq M \qquad [3.43]$$

It is to be noted that the above rules are inadequate as

can easily be imagined when considering what the response of the system will be when it is presented with an utterance that is not among the words in its vocabulary. The system, having evaluated the minimum distance, will assign the unknown word to one of the words in its vocabulary commiting a grave error. A mechanism to guard against such an eventuality needs to be incorporated into the decision strategy. One such protection is to investigate the statistical distance distribution of the utterances of the same word and establish thresholds based on this investigation. Given the threshold for k'th word as S_k, the K-NN rule might be modified such that the classfication to class k is made only if in addition to condition (3.43), $d_k \leq S_k$ and rejects the input utterance as unrecognizable otherwise. Various heuristic rules can be built into the system. Rabiner et al.,[29] discuss an empirically established threshold function in the dynamic time warping stage such that if during the matching process the accumulated distance exceeds the value of the threshold function at that point the matching is abandoned and if this occurs for all the reference templates then the system rejects the input utterance as not belonging to any of stored vocabulary. It must be noted, however, that inspite of such protection the system is liable to make mistakes. The two types of possible errors are rejection and misclassification: the first pertains to the situation where the system rejects a valid utterance because it fails to satisfy the threshold condition and the latter refers to the

case when the system identifies an utterance as belonging to an invalid class because it satisfies all the classification conditions.

The system might also be designed to only report the first L nearest possible candidate classes and the recognition task performed by an upper level subsystem that incorporates higher-level strategy to perform the classification. Systems of this type are the speech understanding systems that require a some sort of speech recognizer as an interface to the acoustic signals.

3.8 Summary

In this chapter, the isolated word recognition system structure was reviewed. Four components were identified and presented in some detail. The concepts of distance measure was presented and the registration of two different utterances explained. The clustering requirements were shown to be needed in cases were the recognition is to be speaker independent. Finally, a discussion of the decsion strategy was presented.

CHAPTER IV
DISCUSSION OF EXPERIMENTAL IMPELEMENTATION

4.1 INTRODUCTION

This chapter will be devoted to a description of the experimental implementation of the speaker independent Arabic digits recognition system. The parameters used for the four architectural components as discussed in chapter III and results obtained at each stage will be presented. Next the performance of the system is analysed in each of the three operating modes-training, clustering and testing. The FFS and VFS are compared with respect to ease of implementation, storage and computer time requirements.

4.2 DATA AQUISITION

4.2.1 ANALOG DATA COLLECTION

Four utterances for each word from twenty speakers were recorded in a normal laboratory environment using a Sony stereo cassette recorder and were recorded on metal cassettes of frequency response in the range 60-13000Hz. The samples were spoken by speakers of varied accents and nationalities in order to include, as much as possible, various accents and dialects. It was decided to use colloquial Arabic at the expense of increased variance among utterances. The 800 utterances from the twenty speakers

were used to train the system. Another batch of test utterances were also recorded under the same conditions.

4.2.2 Digitization

The anolog utterances were band-pass filtered between 80Hz-3500Hz. The filtered signal was then digitized at 8KHz using a Tecmar PC-Master A/D converter installed on an IBM-PC XT at 12 bits/sample. A 2.5s portion of each utterance imbedded in silence was first sampled to obtain 20000 samples. The sampled speech signals were played back using a D/A converter that was connected to a loud speaker and a sliding window of length 8192 samples (1.25s) was then used to zoom on the required speech utterance. This semi-automatic procedure was a first step in the endpoint determination of the utterances. Figure 18 is a block diagram representation of the data aquisition set up.

4.2.3 Data Transfer

Because the recognition system was implemented on the IBM 3033, the data was transfered from the IBM-PC to the mainframe using the IRMA interface card between the two. A program was written to handle this data transfer as the IRMA data transfer software was found to be inadequate for binary data transfer both in speed and in maintaining the data integrity. A data transfer rate of one utterance of 8192 samples in 45 seconds was the maximum rate achieved.

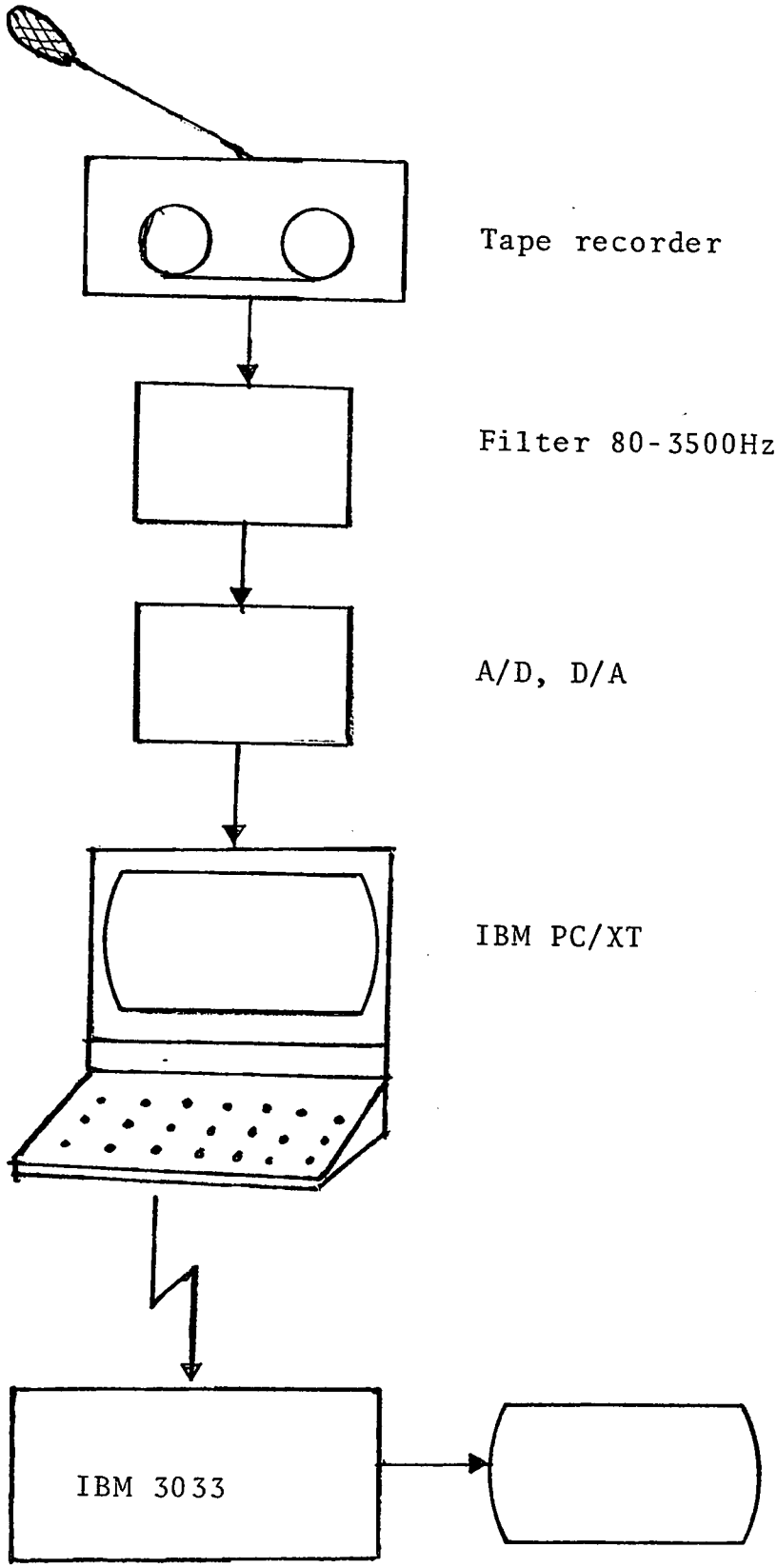


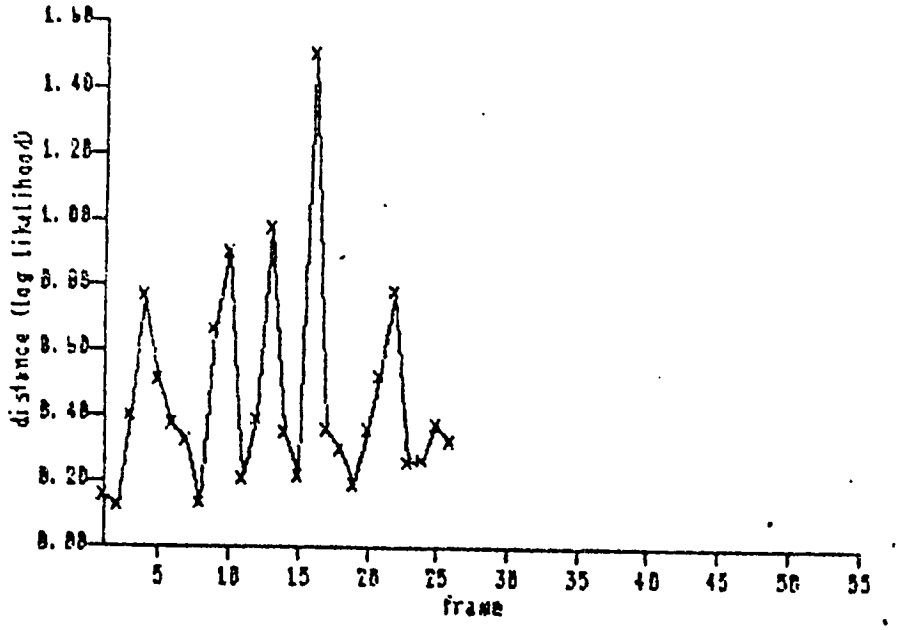
Figure 18: The data aquisition set up.

4.3 PREPROCESSING

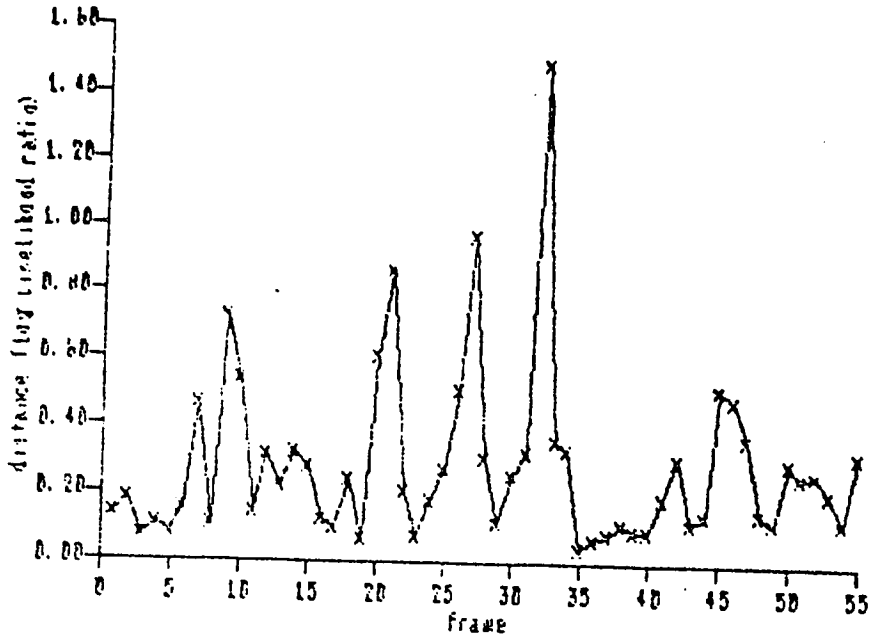
The preprocessing required for the system were endpoint detection, premphasis, framing and windowing. The automatic endpoint detection algorithm given in [15] was used. All utterances were pre-emphasized using the filter $P(z)=1-.95*Z^{-1}$. Two framing approaches were implemented; (a) frames of fixed size and (b) frames of variable size. In both cases, the model order was selected to be 12. The Hamming window of appropriate length as presented in chapter III was applied to both the fixed size and variable size frames.

The fixed size frames (FFS) were of 256 points (32ms) with frame overlap of 128 (16ms). In implementing the frames of variable size (VFS), first a fixed size frame was formed, the required preprocessing performed and a complete LPC analysis conducted. The distance between the current and the previous frame was calculated and if found to be less than the preset threshold value, likelihood ratio of 1.4, the two frames were merged and treated as one frame. A Hamming window was then performed on the merged frame and the LPC parameters extracted. The procedure was repeated and up to three consecutive frames were allowed to be merged. A flow graph of the merging procedure is given in Appendix C.

Figure 19 illustrates the distances between successive frames when an Arabic utterance of /sifr/ was blocked at both fixed and variable sizes. The minima in the graph represent frame positions were the previous and the current



(b) distance between successive frames of the compressed word sifr



(a) distance between successive frames of the word sifr

Figure 19: The distance between successive frames of the Arabic word sifr a) the orginal word and b) after merging.

frames are close enough to be considered for merging. Unlike the situation we implemented, if there is no upper limit on the number of consecutive frames that can be merged, the minima in the interframe distance graph would all exceed the merging threshold. Moreover, some of the minima in the original graph disappear if the time compressed frames are such that the remaining neighbouring frames are far away from one another. The figure further shows the dramatic decrease in the number of frames per utterance when variable frame coding was applied.

4.4 Feature Extraction

Because the primary use of the features is to enable utterances to be compared both in the clustering and the classification phases, a decisive factor in determining as to what form the stored features assume is the mathematical form of the distance function used. Any form that introduces computational simplifications is more desirable because of the substantial computational saving that would be realized in the classification stage when the unknown input utterance is dynamically time-warped with all the stored templates.

It was shown in chapter III that the log likelihood ratio distance between two frames whose LPC are given by by **A** and **B** assumes the form

$$d(A,B) = \log [B^t R_x B \; / \; A^t R_x A] \qquad [4.1a]$$

$$= \text{Log} \frac{\left[r_0 g_0 + 2 \sum_{k=1}^{p} r_k g_k \right]}{\left[r_0 f_0 + 2 \sum_{k=1}^{p} r_k f_k \right]} \qquad [4.1b]$$

where f_k is the autocorrelation of the LPC parameters $A^t=[1,a_1,a_2,\ldots,a_p]$ and g_k is the autocorrelation of the LPC parameters $B^t=[1,b_1,b_2,\ldots,b_p]$ given by

$$f_k = \sum_{i=0}^{p-k} a_i a_{i+k} \qquad 0 \le k \le p \qquad [4.2a]$$

$$g_k = \sum_{i=0}^{p-k} b_i b_{i+k} \qquad 0 \le k \le p \qquad [4.2b]$$

Furthermore, it was shown that the quadratic $A^t R_x A$, when R_x and the LPC vector A are derived from the same sequence x_n, represents the least total squared estimation error. Since this error is obtained as a byproduct of the LPC derivation by Durbin's algorithm, there is no need to reevaluate it every time the distance between two frames has to be evaluated when time aligning utterances. Thus, the relevant features to be stored are the autocorrelation coefficients of the data x_n, r_k $0 \le k \le p$, the autocorrlation coefficients of the LPC, f_k $0 \le k \le p$, and the least total squared residual E_p for p=12.

4.5 Operational modes.

Next we present results of the three operational modes of the implemented system. Figure 20 illustrates the software implementation of the recognition system in block diagram form.

4.5.1 Training.

In the implementation of some pattern recognition systems complex learning algorithms are used to process the training patterns. In the case of the isolated word recognition system implemented, however, the only required action is to extract and store the features described previously.

Two files were created, the first to contain the autocorrelation coefficients of the LPC and the least total squared error and the second file to store the data autocorrelation coefficients of all the 800 training samples.

Table 1 presents both the mean number of frames and the standard deviation (truncated to integral values) for both the fixed and the variable size frames. We observe that not only does the number of frames per utterance decreases in varialbe frame coding approach but that the spread is reduced as well.

4.5.2 Clustering

A detailed presentation of the clustering algorithm used in the implementation of the IWRS system was given in chapter III where it was noted that the user-supplied inputs

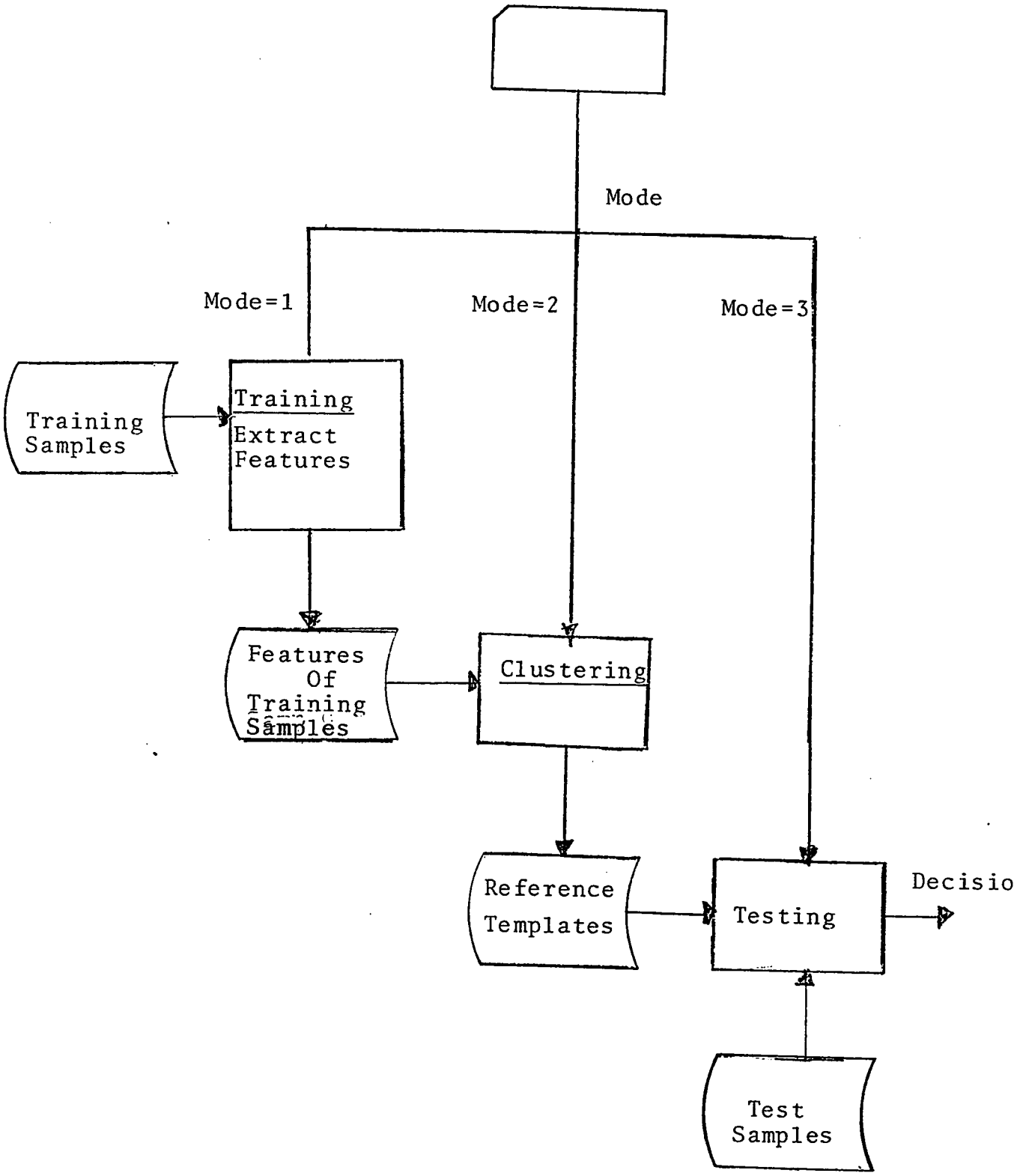


Figure 20: Block Diagram of the software implementation of the recognition system.

Table 1

Mean number of frames and standard deviation for FFS and VFS.

	FFS		VFS	
Word	Mean	Standard deviation	Mean	Standard deviation
sifr	35	8	17	5
wahd	35	10	16	5
ithnen	37	10	16	5
thelatha	41	7	18	4
arbea	37	10	16	6
khemsa	40	10	19	5
sita	39	11	18	4
sebea	38	7	17	4
themania	41	10	18	3
tisa	33	11	16	4

to the algorithm are the distance threshold T and the maximum iteration count KM. The values for the FFS and VFS implementations selected were 0.6 and 0.5 respectively. In both cases a value of KM=5 was used. Table 2 gives results of the clustering algorithm for both the fixed size and variable size frame implementations. The difference between the two are due to different selections of the threshold parameter used. In general inter-utterance distances for the VFS are smaller than those of the FFS method. This arises due to the smaller number of frames for the VFS method. Thus it was necessary to use a smaller threshold parameter for the VFS method which resulted in a different number of cluster centers for the two methods. Therefore, in order to obtain meaningful results when the performance of both methods are compared, the number of clusters used for the VFS implementation were made at most equal to those of the FFS method by discarding the cluster centers with the smallest number of utterances in their cluster set. Table 3 and 4 show the different cluster centers and the number of utterances in each cluster set for the FFS and VFS methods respectively.

The clustering algorithm rejects any cluster center that contains only itself in the cluster set. The number of these utterances, known as outliers, is given in table 5 for both the VFS and FFS implementations.

Table 2

Result of clustering algorithm

	FFS	VFS
Word	Number of clusters	Number of clusters
sifr	11	13
wahd	11	15
ithnen	12	13
thelatha	11	12
arbea	11	12
khemsa	10	12
sita	9	11
sebea	9	11
themania	11	15
tisa	14	11

Table 3

Number of utterances/cluster center for the FFS method

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
sifr	16	16	3	3	7	7	9	5	3	2	3			
wahd	14	6	11	17	4	2	6	5	2	3	3			
ithnen	5	13	7	5	7	6	4	9	8	5	2	3		
thelatha	26	22	3	4	2	4	3	6	2	2	2			
arbea	18	23	4	7	3	7	3	3	2	5	2			
khemsa	43	6	6	6	2	4	3	3	4	2				
sita	7	2	42	7	3	2	4	6	2					
sebea	31	4	14	5	2	3	6	4	3					
themania	25	12	7	3	10	3	3	2	2	5	3			
tisa	10	14	14	4	5	5	2	8	4	2	3	4	2	2

Table 4

Number of utterances/cluster center for the VFS method

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
sifr	8	9	22	5	7	3	4	3	3	3	3			
wahd	13	8	4	10	6	3	3	6	4	4	3			
ithnen	21	4	8	8	8	4	4	3	3	5	3	2		
thelatha	25	7	6	5	4	5	3	8	5	2	4			
arbea	14	5	12	20	3	3	4	4	2	2	6			
khemsa	33	4	6	9	5	3	3	3	2	3				
sita	18	26	3	4	3	5	4	3	3					
sebea	18	9	7	5	4	6	6	9	5	2	2			
themania	10	12	9	6	5	2	5	5	4	6				
tisa	18	12	6	12	3	2	5	3	6	4	2			

Table 5

Number of Outliers for the FFS and VFS methods

	FFS	VFS
Word	Number of outliers	Number of outliers
sifr	6	6
wahd	8	8
ithnen	6	5
thelatha	4	4
arbea	3	3
khemsa	3	5
sita	5	5
sebea	8	7
themania	5	6
tisa	1	7

4.5.3 Testing

The system was tested using three test sources.

1. Of the 800 utterances in the training group, only 109 for the FFS and 105 for the VFS implementations were selected as reference templates. The remaining 691 for the FFS and 695 for the VFS implementations formed the first test group.
2. A second batch of test utterances was created from 12 of the speakers used in the training phase. 120 utterances, one utterance per word from each speaker were collected.
3. A third group of test utterances was formed from 6 speakers not used in the training stage. 2 utterances for a word from five speakers and 3 utterances for a word from the six'th speaker gave a total of 130 test utterances in this batch.

Each of the test utterances were dynamically time warped against all the stored reference templates. The decision strategy adopted was the K-NN rule where K was set to 3. A rejection threshold for each digit was also used and was set equal to the mean of the distance between utterances of the same digit. Table 6 shows the thresholds used for the FFS and VFS methods.

Table 6

Rejection thresholds for the two implementations.

WORD	FFS	VFS
sifr	0.89202720	0.78319907
wahd	0.89902782	0.75429213
ithnen	0.94698453	0.85655814
thelatha	0.82621747	0.70327097
arbea	0.84125143	0.74486995
khemsa	0.80927855	0.67510325
sita	0.75799131	0.69140428
sebea	0.74959373	0.74793702
themania	0.86931318	0.75301939
tisa	0.89476645	0.74842197

The recognition results of the first test type of utterances that were used in the clustering stage but that were not selected as reference patterns is shown in table 7 for both the fixed size and variable size frame implementations. Similarly, tables 8 and 9 give the number of utterances recognized for each digit for test types 2 and 3. A summary of the overall percentage recognition results for the three test types is given in table 10 again for both the fixed size and variable size frame implementations. In tables 11-12 we show the number and of the mis-recognized words and and the classes to which they were assigned.

Table 13-16 show the distances from the cluster centers for representative cases for the VFS and FFS methods for utterances that were recognized and mis-classified. They indicate the cases where distances between utterances of different words can be smaller than those of the same word.

To see the effect of the number of speakers in the training phase, four situations were considered for the FFS; 10 templates for each word from 5, 10, 15 and 20 training speakers were obtained and the systems were tested using test type 3. The results are shown in table 17.

A recognition run of 40 input utterances on the Amdahl 580 for both the FFS and VFS implemenations showed that it took an average of 5.79s and 2.33s per utterance respectively to preprocess, extract the LPC feature vectors and perform dynamic time-warping against all reference templates.

Table 7

Percentage recognition results for test type 1.

WORD	FFS	VFS
sifr	97.10	97.02
wahd	94.20	96.92
ithnen	92.65	91.04
thelatha	93.33	95.59
arbea	94.91	94.12
khemsa	97.14	92.65
sita	92.96	94.20
sebea	92.96	92.75
themania	92.96	95.38
tisa	100.00	85.51

Table 8

Number of utterances recognized out of 12 utterances of test type 2.

WORD	FFS	VFS
sifr	10	9
wahd	11	11
ithnen	11	12
thelatha	11	12
arbea	11	9
khemsa	11	11
sita	12	12
sebea	12	10
themania	11	12
tisa	12	12

Table 9

Number of utterances recognized out of 13 utterances of test type 3.

WORD	FFS	VFS
sifr	10	10
wahd	13	12
ithnen	13	12
thelatha	13	13
arbea	13	12
khemsa	13	13
sita	11	13
sebea	12	12
themania	12	13
tisa	13	13

Table 10

Overall percentage recognition results for the three test types.

	FFS			VFS		
Test Type	recognized	rejected	mis recognized	recognized	rejected	mis recognized
1	94.78	1.19	4.03	93.43	1.94	4.63
2	93.33	0.83	5.84	91.67	1.66	6.66
3	94.62	0.00	5.38	94.62	0.77	4.61

Table 11

Distribution of the mis-recognized utterances for the FFS method.

	number misrecognized	0	1	2	3	4	5	6	7	8	9
sifr	7				2	1			1		3
wahd	2								2		
ithnen	6							1	2	1	2
thelatha	3	1							1		1
arbea	4								7		
khemsa	2					1					1
sita	5	1									4
sebea	6					5					1
themania	5	4							1		
tisa	0										
total	40	6	0	0	2	7	0	1	11	1	12

Table 12.

Distribution of the mis-recognized utterances for the FFS method.

	number misrecognized	0	1	2	3	4	5	6	7	8	9
sifr	7			1	1		1	1	2	1	
wahd	1								1		
ithnen	7	1			2					4	
thelatha	1									1	
arbea	7								7		
khemsa	1								1		
sita	2		1								1
sebea	7				2	4					1
themania	3			1	1				1		
tisa	10	1					1	8			
totla	46	2	0	3	6	4	2	9	12	6	2

Table 13

Distances of a correctly classified utterance (sita) from the cluster centers (FFS).

WORD	1	2	3	4	5	6	7	8	9	10	11	12	13	14
sifr	0.738	0.776	0.883	0.888	0.951	1.059	1.133	1.146	1.314	1.344	1.431			
wahd	0.705	1.127	1.152	1.222	1.253	1.262	1.313	1.333	1.336	1.529	1.613			
ithnen	0.665	0.763	0.838	0.903	0.962	0.980	1.009	1.112	1.172	1.302	1.388	1.544		
thelata	0.733	0.900	0.908	0.989	1.028	1.057	1.069	1.118	1.287	1.354	1.456			
arbea	0.882	0.954	1.085	1.183	1.286	1.340	1.342	1.353	1.405	1.526	1.543			
khemsa	0.509	0.794	0.801	0.806	0.835	0.857	1.050	1.132	1.195	1.397				
sita	0.488	0.544	0.639	0.670	0.741	0.951	1.096	1.181	1.405					
sebea	0.741	0.954	0.996	1.006	1.100	1.131	1.146	1.148	1.234					
themania	0.793	0.834	0.988	1.030	1.059	1.073	1.237	1.279	1.296	1.306	1.464			
tisa	0.544	0.546	0.780	0.858	1.055	1.065	1.101	1.102	1.130	1.132	1.180	1.206	1.278	1.396

Table 14

Distances of a mis-classified utterance (sita) from the cluster centers (FFS).

WORD	1	2	3	4	5	6	7	8	9	10	11	12	13	14
sifr	0.913	0.926	0.939	0.964	1.041	1.078	1.139	1.183	1.244	1.571	1.625			
wahd	1.029	1.042	1.184	1.217	1.289	1.323	1.466	1.470	1.486	1.494	1.538			
ithnen	0.800	0.935	1.058	1.060	1.124	1.163	1.167	1.181	1.194	1.261	1.359	1.392		
thelata	0.889	0.897	1.022	1.030	1.086	1.118	1.138	1.158	1.232	1.294	1.630			
arbea	0.995	1.022	1.063	1.084	1.086	1.100	1.206	1.259	1.263	1.516	1.699			
khemsa	0.674	0.761	0.810	0.817	0.835	0.994	1.081	1.105	1.247	1.749				
sita	0.644	0.713	0.798	0.930	0.942	0.963	0.987	1.131	1.789					
sebea	1.005	1.006	1.053	1.079	1.081	1.101	1.177	1.228	1.264					
themania	0.989	1.000	1.119	1.126	1.139	1.206	1.252	1.357	1.362	1.426	1.459			
tisa	0.415	0.601	0.924	0.925	1.006	1.016	1.080	1.091	1.096	1.148	1.220	1.387	1.409	1.419

Table 15

Distances of a correctly classified utterance (sifr) from the cluster centers (VFS).

WORD	1	2	3	4	5	6	7	8	9	10	11	12
sifr	0.594	0.615	0.641	0.704	0.762	0.794	0.850	0.897	0.933	1.022	1.043	
wahd	0.812	0.884	0.963	1.185	1.244	1.245	1.246	1.364	1.367	1.377	1.608	
ithnen	0.599	0.777	0.845	0.916	0.991	1.017	1.122	1.125	1.276	1.366	1.367	1.726
thelata	0.738	0.776	0.789	0.801	1.011	1.027	1.035	1.211	1.243	1.296	1.300	
arbea	0.861	0.918	1.121	1.160	1.241	1.249	1.266	1.293	1.300	1.451	1.659	
khemsa	0.791	0.913	0.991	1.079	1.156	1.216	1.233	1.256	1.339	1.374		
sita	0.710	0.754	0.764	0.784	0.803	0.847	0.985	1.010	1.040	1.180	1.305	
sebea	0.666	0.723	0.753	0.756	0.763	0.813	0.819	0.929	0.965			
themania	0.564	0.775	0.822	0.835	0.899	0.936	0.957	0.984	1.049	1.104	1.391	
tisa	0.804	0.833	0.883	0.906	0.910	0.918	0.983	1.004	1.036	1.096	1.393	

Table 16

Distances of a mis-classified utterance (sita) from the cluster centers (VFS).

WORD	1	2	3	4	5	6	7	8	9	10	11	12
sifr	0.549	0.818	0.849	0.946	1.035	1.088	1.092	1.108	1.108	1.160	1.620	
wahd	0.770	0.862	0.965	1.030	1.205	1.219	1.232	1.427	1.442	1.529	1.539	
ithnen	0.466	0.713	0.895	0.965	0.969	0.984	1.051	1.077	1.246	1.261	1.383	1.542
thelata	0.681	0.698	0.977	1.006	1.096	1.105	1.127	1.184	1.326	1.331	1.464	
arbea	0.637	0.844	1.043	1.055	1.080	1.194	1.245	1.246	1.252	1.306	1.332	
khemsa	0.896	0.903	1.049	1.072	1.290	1.320	1.342	1.342	1.350	1.551		
sita	0.801	0.815	1.021	1.088	1.095	1.122	1.144	1.159	1.194	1.214	1.298	
sebea	0.545	0.879	0.919	1.017	1.019	1.037	1.143	1.147	1.261			
themania	0.555	0.707	0.871	1.086	1.088	1.105	1.120	1.162	1.181	1.227	1.298	
tisa	0.745	0.987	1.034	1.129	1.193	1.229	1.243	1.244	1.261	1.350	1.692	

Table 17

Overall percentage recognition results for Test Type 3 for variable number of speakers and templates in the training phase.

number of speakers	average number of templates	recognized	rejected	mis recognized
5	5	83.08	7.69	9.23
10	7	85.38	1.54	13.08
15	8	89.23	0.77	10.00
20	10	94.62	0.00	5.38

Table 18

Overall percentage recognition results for Test Type 3 for variable number of speakers and but fixed number of templates in the training stage.

number of speakers	average number of templates	recognized	rejected	mis recognized
5	10	93.08	0.00	6.92
10	10	91.54	1.54	6.92
15	10	93.85	0.77	5.38
20	10	94.62	0.00	5.38

4.6 Discussion of Results

A comparison of the two methods of framing, fixed and variable, from table 10 shows that the former method is consistently superior in recognition performance when the overall recognition rates are compared. However, the results for the test type 1, those utterances used in the training stage but that were not used as cluster centers, as given in table 7 reveals that recognition improves the VFS method in some cases while there is a marked degradation in other cases. While the FFS method resulted in a perfect score for the word /tisea/, the case for the VFS method was the poorest. It seems that the merging of frames in cases like /wahd/, /thelatha/ and /themania/ removes redundant information thus improving the recognition rates for those words, while it results in the loss of valuable information which manifests itself severly in the case of words like /khemsa/ and /tisea/.

This inconsistency can be attributed to the distance function upon which the merging of neighbouring frames is based upon. The Itakura measure, being a probabilitic measure, does not guarantee, in all cases, that low distance values evaluated necessarily imply the similarity of the sound sources generating the frames.

Tables 11-12 indicate the number of times a mis-recognized word is classified to another member of the vocabulary. Though it is difficult to interpret this results, nonetheless, some comments can be made. No words were mis-recognized as /wahd/,/ithnen/ and /thelatha/ for

FFS method while only one utterance was mis-recognized as /wahd/ for the VFS method. For both cases all the mis-recognized utterances for /arbea/ were classified as /sebea/ while the majority of the mis-recognized utterances of /sebea/ were attributed to /arbea/ indicating a strong similarity between these two words as far as the method of recognition implemented is concerned. Such a similarity also can be seen for /sita/ and /tisea/. The majority of the utterances mis-recognized were classified to /sita/, /sebea/ and /themania/ for the VFS method while they were classified to /tisea/, /sebea/ and /arbea/ in the FFS method.

Figures 21a and 21b show the distance distribution from the reference templates of /sita/ of the training samples of /sita/ and of the training samples of /tisea/ for the FFS and VFS methods respectively. The figures demonstrate the overlap between the two words which for the VFS method is severe explaining the recognition degradation for /tisea/ in the VFS method.

Figures 22a and 22b show the distance distribution from the reference templates of /arbea/ of the training samples of /arbea/ and of the training samples of /sebea/ and from the reference templates of /arbea/ of the training samples of /arbea/ and of the training samples of /sita/ respectively. The figures clearly show the strong similarity between /arbea/ and /sebea/ and the the strong dissimilarity between /arbea/ and /sita/.

Finally, figures 23a and 23b show the distance distribution from the reference templates of /khemsa/ of the training samples of /khemsa/ and of the reference templates of all the other words and from the reference templates of /tisea/ of the training samples of /tisea/ and of the reference templates of all the other words respectively for the FFS method. The relatively small overlap in (a) and the strong overlap in (b) helps to explain why none of the mis-recognized utterances were classified to /khemsa/ and the reason the majority of the mis-classified utterances were classified as /tisea/ as given in table 11.

Table 17 and 18 show the overall recognition rates for test type 3 for the number of speakers in training phase of 5, 10, 15 20 and when the number of stored templates were variable and fixed respectively. The steady increase in the recognition rate shown in table 17 can be attributed not to the increase in the number of speakers but to the increase in the number of stored reference templates as is evident from table 18. Although the test samples of only 130 utterances may not be sufficient to indicate the effect of increasing the number of speakers on the recognition rates, the results are however significant.

The recognition rate for both cases is consistent with those reported in the literature which range from a low of 85% to 99% with the higher rates bing for speaker-dependent implementations. Other factors that affect the recogntion rates are the recording environment, the nature and size of the vocabulary, the features used and the number of

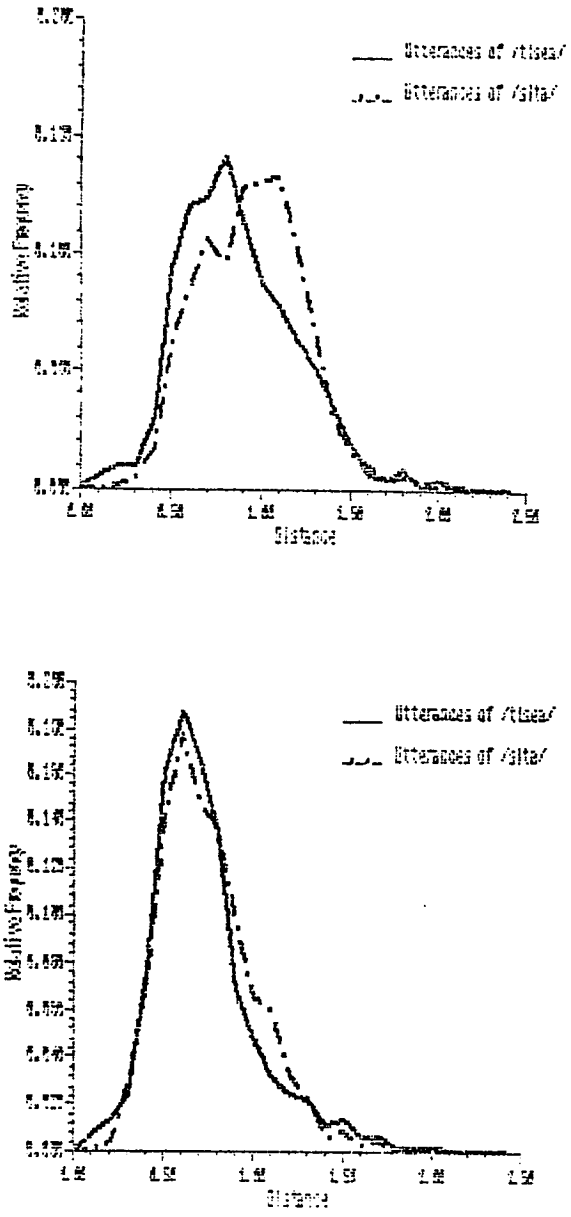
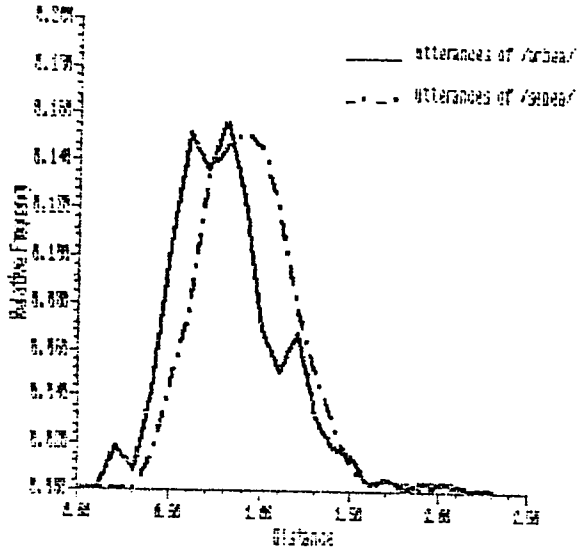


Figure 21: Distance Distribution from the reference templates of /sita/ of the training samples of /sita/ and /tisea/. (a) FFS method. (b) VFS method.

(a)



(b)

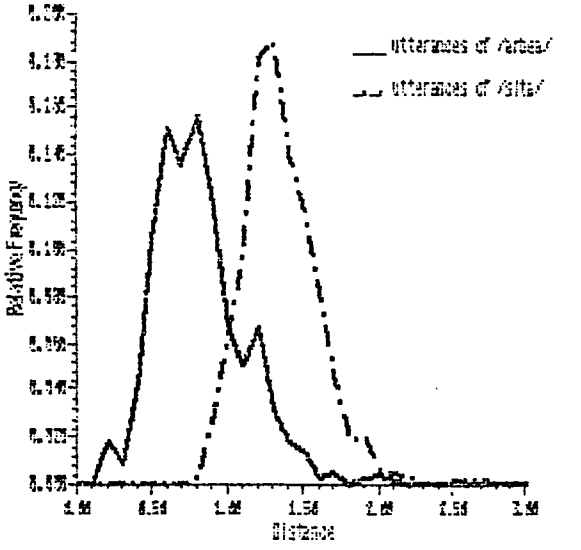
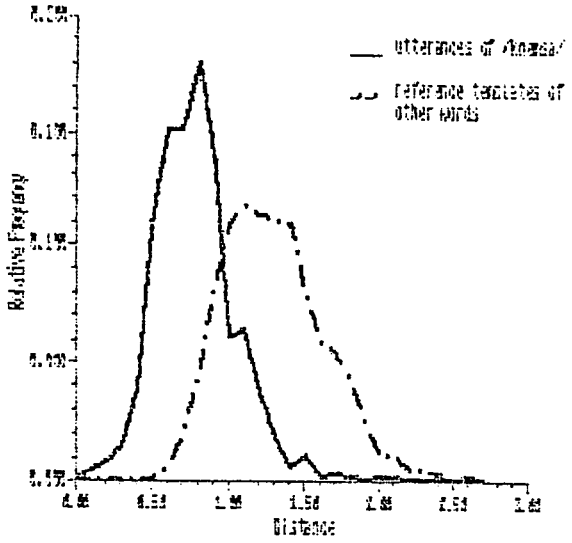


Figure 22: Distance Distribution from reference templates of /arbea/. (a) of the training samples of /arbea/ and training samples of /sebea/. (b) of the training samples of /arbea/ and training samples of /sita/.

(a)



(b)

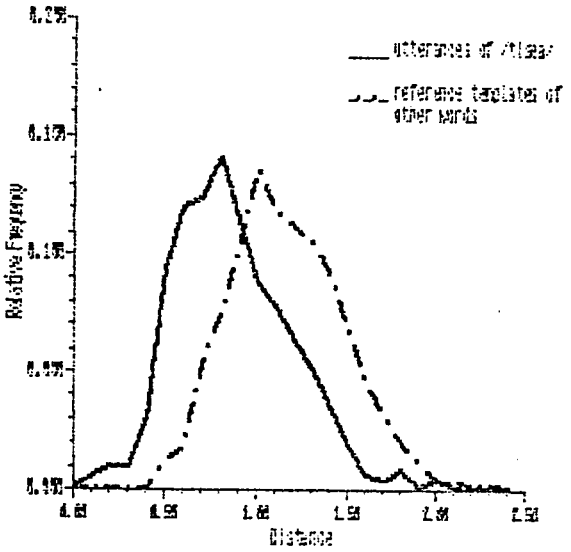


Figure 23: Distance Distribution from the reference templates of two words. (a) from /khemsa/ of the training samples of /kehemsa/ and all reference templates of the remaining words. (b) from /tisea/ of the training samples of /tisea/ and all reference templates of the remaining words.

heuristic rules incorporated in the system. It is evident from tables 14 and 16 that the distance of an utterance from cluster centers of a different word may be smaller than those from the actual word.

The imperfect performance of the isolated word recognition systems is partly due to the dynamic programming technique of distance evaluation which assigns equal weight to all interframe dissimilarities which creates errors when the words in the vocabulary have phonemically similar parts, which is the case for instance between the Arabic word /arbea/ (four) and /sebea/ (seven) whose last phenomes are identical. The significant differences between these words is in first phenomes /s/ and /a/. By obtaining a global optimal path within the search space, the dynamic programming doesnot emphasize those aspects which differentiate the utterances but rather works uniformly. A further source of error is the unavoidable diversity in the pronounciations of the words used which degrades perfomance when the system is operated in the speaker independent mode.

A technique that has been attempted to overcome the shortcoming of the dynamic programming technique is the two-pass approach [55] wherein the vocabulary is divided into similar sounding subclasses and the recognition procedure determines the subclass of the input utterance in the first pass and attempts final recognition within the identified subclass in the second pass. It is evident that these procedure, while improving recognition performance, is time consuming and impractical for real-time implementations.

Another limitation is that the method of multiple templates becomes impractical as the number of the words in the vocabulary is increased. Thus, although the method works reasonably for small size vocabularies, it is necessary to use minimal representations, such as phonemes, in the case of medium sized and large vocabulary systems.

On the average, it took more than twice computing time for the FFS method to preprocess, extract the LPC features and template match an input utterance with the stored references. This difference arises from the reduced number of frames for the VFS method where it is seen, from table 1, on the average the number of frames for the VFS method is less than half of that for the FFS method.

4.7 Conclusion

The discussion so far has demonstrated that it is possible to implement a small vacabulary speaker independent recognition system based on LPC features and reference templates obtained using clustering techniques. The technique has some disadvantages, however. Prominent among these is the necessity of maintaining multiple reference templates for a word which rapidly renders the method impractical as the size of the vocabulary increases.

Some confusability was detected between the words /arbea/ and /sebea/ and between the words /sita/ and /tisea/. In general, once such sets of confusable words are identified, methods must be devised to enhance the features that discriminate between them.

Using a general purpose computer to template match an input utterance with all stored templates, although possible for batch processing as was implemented, is impractical for real-time processing. Thus a method of compressing the speech data is a must although such a compression may result in relevant information being thrown away. The technique of variable frame size coding, while improving the recognition speed and reducing the size of the information to be stored, offers inconsistent perfomance improvement in term of recognition results. The loss of valuable information leads to unacceptable recognition degradation in some cases while some improvement was registered in other cases suggesting that it should be used with care if time-compression is being considered.

It was observed that the recognition results varied insignificantly as the number of speakers in the training phase were increased when an equal number of templates were used. It can be concluded that the significant factor to improve speaker-independent performance is the number of different templates used.

4.8 Recommendations for Further Research.

In the following discussion we present some ideas for further research in the area of speaker independent isolated word recognition systems.

1. The primary method used to obtain speaker independence in this work was the method of statistical clustering which necessitated the use of multiple reference templates to be stored for each word of the vocabulary. It is an ideal situation if only one reference template could be sufficient to represent a word. Thus a legitimate field of research is the investigation of a learning procedure in which during the training stage the recognition system uses the input utterances to modify the reference template for that word by minimizing some function of the distances between the modified reference template and all the training utterances used.

2. It has been mentioned that the dynamic time warping algorithm assigns equal weight to all interframe distances between two frames of the two utterances being compared. In particular, no consideration of the variation within the frames of the same utterance is taken. The suggestion is to investigate the effect of various weighting functions of the distances of the currently considered frames of the two utterances from their respective previously matched frames on the performance of the recognition system.

3. A third suggestion is to perform the dynamic time warping algorithm in a piecewise fashion. The utterances are divided into a predetermined number of parts, N, where these parts are of variable time-duration. The variable time-duration parts may be obtained, for instance, by evaluating the interframe distances and assigning the boundary of the N parts as those times where the interframe distances where biggest. The N parts of the two utterances are aligned and the dynamic programming optimization peformed in between. The distance between the utterances is then evaluated as the sum of the individual distances.

APPENDIX A
DERIVATION OF THE PARTIAL CORRELATION COEFFICIENTS.

In order to derive the partial correlation coefficients, we digress here to introduce the concept of forward and backward predictions. Given the sequence x_n, we can define two predictions, a forward prediction and a backward prediction.

$$x'_n = - \sum_{k=1}^{m} a_k^m x_{n-k} \qquad [A.1a]$$

$$x'_{n-m-1} = - \sum_{k=1}^{m} b_k^m x_{n-m-1+k} \qquad [A.1b]$$

where $\{a_k^m\}$, $\{b_k^m\}$ are the coefficients of the forward and the backward predictions respectively shown in figure 23. These two estimates result in a backward and a forward error sequences as

$$\varepsilon_m^+(n) = x_n - x'_n$$

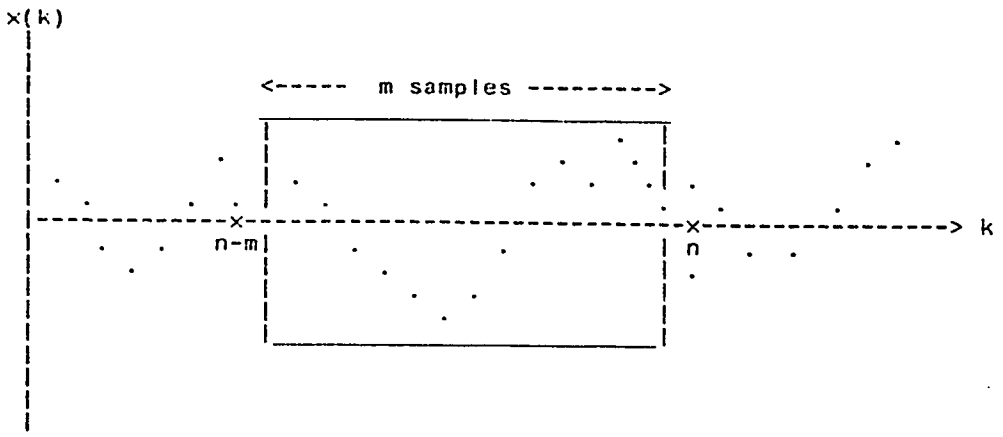


Figure 24: forward and backward prediction

$$= x_n + \sum_{k=1}^{m} a_k^m \, x_{n-k}$$

$$= \sum_{k=0}^{m} a_k^m \, x_{n-k} \qquad a_0^m = 1 \qquad \text{[A.2a]}$$

and

$$\bar{\varepsilon}_m(n) = x_{n-m-1} - x'_{n-m-1}$$

$$= x_{n-m-1} + \sum_{k=1}^{m} b_k^m \, x_{n-m-1+k}$$

$$= \sum_{k=0}^{m} b_k^m \, x_{n-m-1+k} \qquad b_0^m = 1 \qquad \text{[A.2b]}$$

where ε^+ and ε^- indicate the forward and backward error sequences respectively and the superscript m indicates the model order. We also form the total backward and forward residuals as

$$E_m^+ = \sum_{n=-\infty}^{\infty} [\varepsilon_m^+(n)]^2 \qquad [A.3a]$$

$$E_m^- = \sum_{n=-\infty}^{\infty} [\varepsilon_m^-(n)]^2 \qquad [A.3b]$$

Minimizing E_m^+ with respect to a_k^m and E_m^- with respect to b_k^m yields the respective m normal equations

$$\sum_{n=-\infty}^{\infty} \varepsilon_m^+(n)\, x_{n-k} = 0 \qquad 1 \le k \le m \qquad [A.4a]$$

$$\sum_{n=-\infty}^{\infty} \varepsilon_m^-(n)\, x_{n-m-1+k} = 0 \qquad 1 \le k \le m \qquad [A.4b]$$

which are the orthogonality conditions previously derived in chapter II.

By inserting for $\varepsilon_m^+(n)$ from (A.2a) into (A.4a) a simplified statement of this orthogonality conditions is obtained as

$$\sum_{n=-\infty}^{\infty} \left[x_n + \sum_{j=1}^{m} a_j^m x_{n-j} \right] x_{n-k} = 0 \qquad [A.5a]$$

$$\sum_{n=-\infty}^{\infty} x_n \, x_{n-k} + \sum_{j=1}^{m} a_j^m \sum_{n=-\infty}^{\infty} x_{n-j} \, x_{n-k} = 0 \qquad [A.5b]$$

$$r_k + \sum_{j=1}^{m} a_j^m \, r_{|j-k|} = 0 \qquad 1 \le k \le m \qquad [A.5c]$$

In the same manner, by replacing for $\varepsilon_m^-(n)$ from (A.2b) from (A.4b) a simpler set of equations of the backward prediction paramters of order m is derived as.

$$\sum_{n=-\infty}^{\infty} [x_{n-m-1} + \sum_{j=1}^{m} b_j^m \, x_{n-m-1+j} \,] x_{n-m-1+k} = 0$$

$$\sum_{n=-\infty}^{\infty} x_{n-m-1} x_{n-m-1+k} + \sum_{j=1}^{m} b_j^m \sum_{n=-\infty}^{\infty} x_{n-m-1+j} x_{n-m-1+k} = 0$$

$$r_k + \sum_{j=1}^{m} b_j^m \, r_{|j-k|} = 0 \qquad 1 \le k \le m \qquad [A.6a]$$

Because the forward and backward linear prediction coefficients $\{a_k^m\}$, $\{b_k^m\}$ satisfy the same set of orthogonality conditions (A.5c) and (A.6) it can be concluded that

$$a_k^m = b_k^m \qquad 1 \le k \le m \qquad [A.7]$$

This is a consequence of the symmetry of the autocorrelation coefficients, mainly $r_k = r_{-k}$ for $1 \le k \le m$

Since by (A.4a) the error sequence $\varepsilon_m^+(n)$ is orthogonal to each of the past m values $x_{n-1},\ldots,x_{n-m}$ and since for any $k<m$ the backward error sequence $\varepsilon_k^-(n)$ of order k is a linear combination of a subset of them, mainly the k+1 values, $x_{n-k-1},x_{n-k},\ldots,x_{n-1}$, it follows that

$$\sum_{n=-\infty}^{\infty} \varepsilon_m^+(n)\, \varepsilon_k^-(n) = 0, \qquad 1 \le k < m \qquad \text{[A.8a]}$$

and when k is equal to m, the backward error sequence $\varepsilon_m^-(n)$ being a linear combination of the past m+1 values $x_{n-1},x_{n-2},\ldots,x_{n-m},x_{n-m-1}$, satisfies

$$\sum_{n=-\infty}^{\infty} \varepsilon_m^-(n)\, \varepsilon_m^+(n) = \sum_{n=-\infty}^{\infty} \varepsilon_m^-(n)\, x_n \qquad \text{[A.8b]}$$

Equation (A.8) say that the forward error sequence $\varepsilon_m^+(n)$ of a filter of order m is uncorrelated with the backward error sequences $\varepsilon_k^-(n)$ of a lesser order filter. A similar line of reasoning leads to the relation

$$\sum_{n=-\infty}^{\infty} \varepsilon_m^-(n)\, \varepsilon_k^-(n) = 0, \qquad 1 \le k < m \qquad \text{[A.9]}$$

Now, because the forward error sequence $\varepsilon_m^+(n)$ is a linear combination of $x_n,x_{n-1},\ldots,x_{n-m}$, and all backward error sequences of order $k<m$ are linear combinations of a subset

of $x_{n-1},\ldots,x_{n-m}$, then $\varepsilon_m^+(n)$ can be expressed as a sum of x_n and the m backward errors $\varepsilon_k^-(n)$ k=0,1,..,m-1, namely

$$\varepsilon_m^+(n) = x_n + \sum_{k=1}^{m} K_k \, \varepsilon_{k-1}^-(n) \qquad [A.10]$$

By comparing this equation with equation (A.2a), namely

$$\varepsilon_m^+(n) = x_n + \sum_{k=1}^{m} a_k^m \, x_{n-k}$$

and noting that in equation (A.10) the only backward error term that contains the term x_{n-m} is $\varepsilon_{m-1}^-(n)$ with coefficient K_m, we are able to conclude that K_m is identical to a_k^m. The values of K_i may be evaluated by multiplying both sides of (A.10) by $\varepsilon_{i-1}^-(n)$ and summing over all the data points to obtain

$$\sum_{n=-\infty}^{\infty} \varepsilon_m^+(n) \, \varepsilon_{i-1}^-(n) = \sum_{n=-\infty}^{\infty} \varepsilon_{i-1}^-(n) x_n + \sum_{k=1}^{m} K_k \sum_{n=-\infty}^{\infty} \varepsilon_{k-1}^-(n) \varepsilon_{i-1}^-(n)$$

$$1 \le i \le m \qquad [A.11]$$

which after applying the relationships derived in (A.8a) and (A.9) reduces to

$$0 = \sum_{n=-\infty}^{\infty} \varepsilon_{i-1}^-(n) \, x_n + K_i \sum_{n=-\infty}^{\infty} \varepsilon_{i-1}^-(n) \, \varepsilon_{i-1}^-(n)$$

$$1 \le i \le m \qquad [A.12]$$

A further simplification is achieved when we apply (A.8b) to

the first term in the right hand side to get

$$0 = \sum_{n=-\infty}^{\infty} \varepsilon_{i-1}^{+}(n)\, \varepsilon_{i-1}^{-}(n) + K_i \sum_{n=-\infty}^{\infty} \varepsilon_{i-1}^{-}(n)\, \varepsilon_{i-1}^{-}(n)$$

$$1 \le i \le m \qquad \text{[A.13]}$$

In particular, when i equals m, we obtain K_m as

$$K_m = - \frac{\sum_{n=-\infty}^{\infty} \varepsilon_{m-1}^{+}(n)\, \varepsilon_{m-1}^{-}(n)}{\sum_{n=-\infty}^{\infty} \varepsilon_{m-1}^{-}(n)\, \varepsilon_{m-1}^{-}(n)} \qquad \text{[A.14]}$$

The denominator in (A.14) is the total backward prediction residual energy of model order m-1 which in the notation used previously is E_{m-1}^{-}. In general, the total backward prediction error E_m^{-} is equal to the total forward prediction error of the same order E_m^{+}. Specifically,

$$E_m^{-} = \sum_{n=-\infty}^{\infty} [\varepsilon_m^{-}(n)]^2 \qquad \text{[A.15a]}$$

$$= \sum_{n=-\infty}^{\infty} \varepsilon_m^{-}(n) \left[x_{n-m-1} + \sum_{k=1}^{m} b_k^m\, x_{n-m-1+k} \right] \qquad \text{[A.15b]}$$

which upon using the orthogonality conditions (A.4b) becomes

$$= \sum_{n=-\infty}^{\infty} \varepsilon_m^-(n)\, x_{n-m-1} \qquad [A.15c]$$

$$= \sum_{n=-\infty}^{\infty} \left[x_{n-m-1} + \sum_{k=1}^{m} b_k^m\, x_{n-m-1+k} \right] x_{n-m-1} \qquad [A.15d]$$

$$= \sum_{k=0}^{m} b_k^m \sum_{n=-\infty}^{\infty} x_{n-m-1+k}\, x_{n-m-1} \qquad [A.15e]$$

$$= \sum_{k=0}^{m} b_k^m\, r_k \qquad [A.15f]$$

Similarly, a parallel derivation shows

$$E_m^+ = \sum_{k=0}^{m} a_k^m\, r_k \qquad [A.16]$$

Because $b_k^m = a_k^m$ for $1 \le k \le m$, it immediately follows from (A.15) and (A.16)

$$E_m^- = E_m^+ \qquad [A.17]$$

that is the forward and backward residual energies are equal which is again a consequence of the symmetry of the autocorrelation coefficients. The numerator in (A.14) is given by

$$\sum_{n=-\infty}^{\infty} \varepsilon_{m-1}^{-}(n)\, x_n \qquad [A.18a]$$

$$= \sum_{n=-\infty}^{\infty} \Big[\sum_{k=0}^{m-1} b_k^{m-1}\, x_{n-m+k} \Big]\, x_n \qquad [A.18b]$$

$$= \sum_{k=0}^{m-1} b_k^{m-1} \Big[\sum_{n=-\infty}^{\infty} x_{n-m+k}\, x_n \Big] \qquad [A.18c]$$

$$= \sum_{k=0}^{m-1} b_k^{m-1}\, r_{m-k} \qquad [A.18d]$$

$$= r_m + \sum_{k=1}^{m-1} b_k^{m-1}\, r_{m-k} \qquad [A.18e]$$

From the equality of the backward and forward residual energies the partial correlation coefficient can now be put as

$$K_m = - \frac{\sum_{n=-\infty}^{\infty} \varepsilon_{m-1}^{+}(n)\, \varepsilon_{m-1}^{-}(n)}{\Big[\sum_{n=-\infty}^{\infty} \varepsilon_{m-1}^{-}(n)\, \varepsilon_{m-1}^{-}(n) \sum_{n=-\infty}^{\infty} \varepsilon_{m-1}^{+}(n)\, \varepsilon_{m-1}^{+}(n) \Big]^{\frac{1}{2}}} \qquad [A.19]$$

which is seen to be the negative correlation coefficient between the residual sequences $\varepsilon_{m-1}^{+}(n)$ and $\varepsilon_{m-1}^{-}(n)$. Since

the error sequences $\varepsilon^{+}_{m-1}(n)$ and $\varepsilon^{-}_{m-1}(n)$ are the result of removing from the sequences x_n abd x_{n-m} the predicted values x'_n and x'_{n-m} respectively, equation (A.19) is interpreted to be the negative of the correlation coefficient between the signal sequences x_n and x_{n-m} after the effect of the intermediate values $x_{n-1}, x_{n-2}, \ldots, x_{n-m+1}$ have been removed from both of them, hence the term partial correlation coefficient.

The total residual energy for the filter of order m can also be generated recursively from that of order m-1 at each step enabling us to avoid the use of equation (A.16). Expressing (A.10) for m-1 order model gives

$$\varepsilon^{+}_{m-1}(n) = x_n + \sum_{k=1}^{m-1} K_k \, \varepsilon^{-}_{k-1}(n) \qquad [A.20]$$

which along with (A.10) immediately produces the expression for error sample as

$$\varepsilon^{+}_{m}(n) = \varepsilon^{+}_{m-1}(n) + K_m \, \varepsilon^{-}_{m-1}(n) \qquad [A.21]$$

This leads to the recursive method as

$$E^{+}_{m} = \sum_{n=-\infty}^{\infty} [\varepsilon^{+}_{m}(n)]^2 \qquad [A.22a]$$

$$= \sum_{n=-\infty}^{\infty} [\varepsilon_{m-1}^{+}(n)]^2 + \sum_{n=-\infty}^{\infty} [\, K_m \varepsilon_{m-1}^{-}(n)\,]^2 \qquad [A.22b]$$

$$+ 2K_m \sum_{n=-\infty}^{\infty} [\varepsilon_{m-1}^{+}(n)\ \varepsilon_{m-1}^{-}(n)]$$

$$= E_{m-1}^{+} + K_m^{\,2}\, E_{m-1}^{-} + 2K_m \sum_{n=-\infty}^{\infty} [\varepsilon_{m-1}^{+}(n)\ \varepsilon_{m-1}^{-}(n)] \qquad [A.22c]$$

The third term of this expression is from (A.19) given by

$$\sum_{n=-\infty}^{\infty} [\varepsilon_{m-1}^{+}(n)\ \varepsilon_{m-1}^{-}(n)] = -\, K_m\, E_{m-1}^{-} \qquad [A.22d]$$

Thus by using this value, (A.22c) becomes

$$= E_{m-1}^{+} + K_m^{\,2}\, E_{m-1}^{-} - 2K_m^{\,2}\, E_{m-1}^{-} \qquad [A.22e]$$

and by the equality of the forward and backward total residuals reduces to

$$E_m^{+} = (1-K_m^{\,2})\, E_{m-1}^{+} \qquad [A.22f]$$

We have covered all the steps required to recursively derive the solutions of the LPC parameters for order m given the solution of order m-1. Specficially, combining (A.14), (A.17) and (A.18e) gives

$$K_m = -\, [r_m + \sum_{k=1}^{m-1} b_k^{m-1}\, r_{m-k}] \,/\, E_{m-1}^{+} \qquad [A.23]$$

Thus we have a method for generating the values K_m at each step which enables the recursive solution to be complete. The only thing remaining to be specified is the initial values of E_0^+. It can be easily seen that the optimal zero order filter is given by

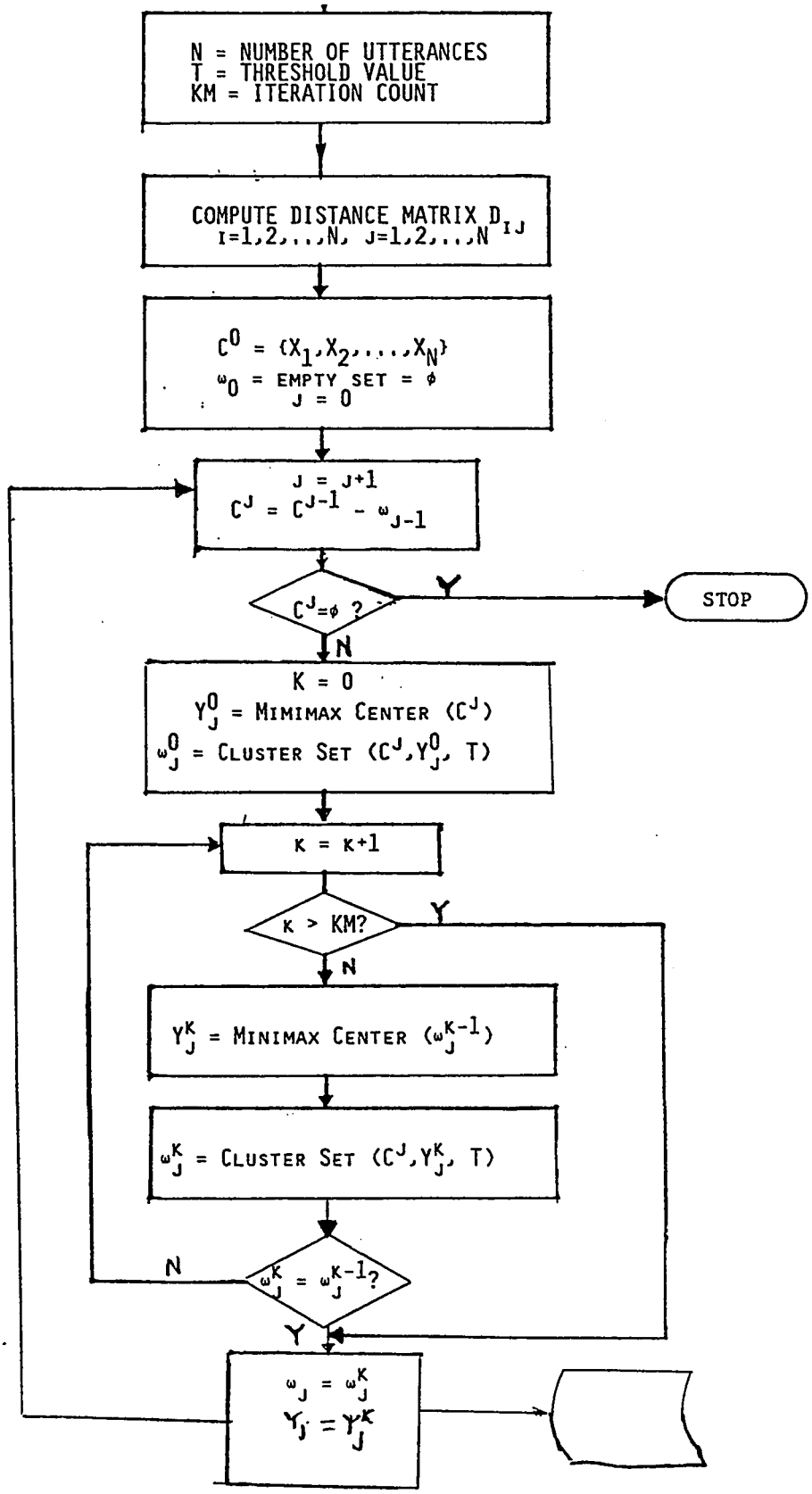
$$a_0^0 = 1 \qquad \text{[A.24a]}$$

and by (A.16) the corresponding total residual at step zero is

$$E_0 = r_0 \qquad \text{[A.24b]}$$

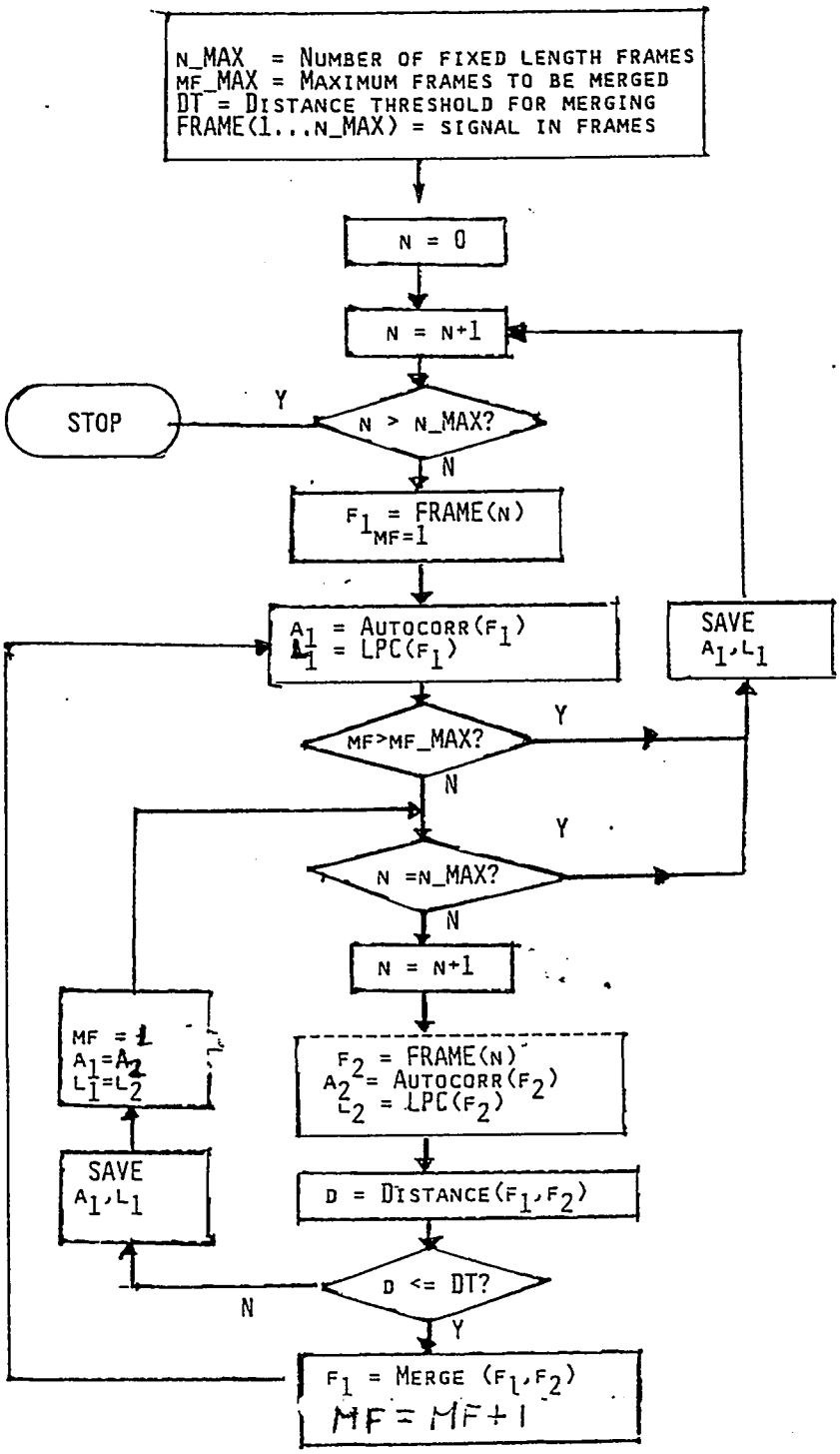
APPENDIX B

THE CLUSTERING ALGORITHM


N = NUMBER OF UTTERANCES
T = THRESHOLD VALUE
KM = ITERATION COUNT
COMPUTE DISTANCE MATRIX D_{IJ} I=1,2,...,N, J=1,2,...,N
$C^0 = (X_1, X_2, \ldots, X_N)$
ω_0 = EMPTY SET = ϕ
J = 0
J = J+1
$C^J = C^{J-1} - \omega_{J-1}$
$C^J = \phi$?
Y
STOP
N
K = 0
Y_J^0 = MIMIMAX CENTER (C^J)
ω_J^0 = CLUSTER SET (C^J, Y_J^0, T)
K = K+1
K > KM?
Y
N
Y_J^K = MINIMAX CENTER (ω_J^{K-1})
ω_J^K = CLUSTER SET (C^J, Y_J^K, T)
$\omega_J^K = \omega_J^{K-1}$?
N
Y
$\omega_J = \omega_J^K$
$Y_J = Y_J^K$

APPENDIX C

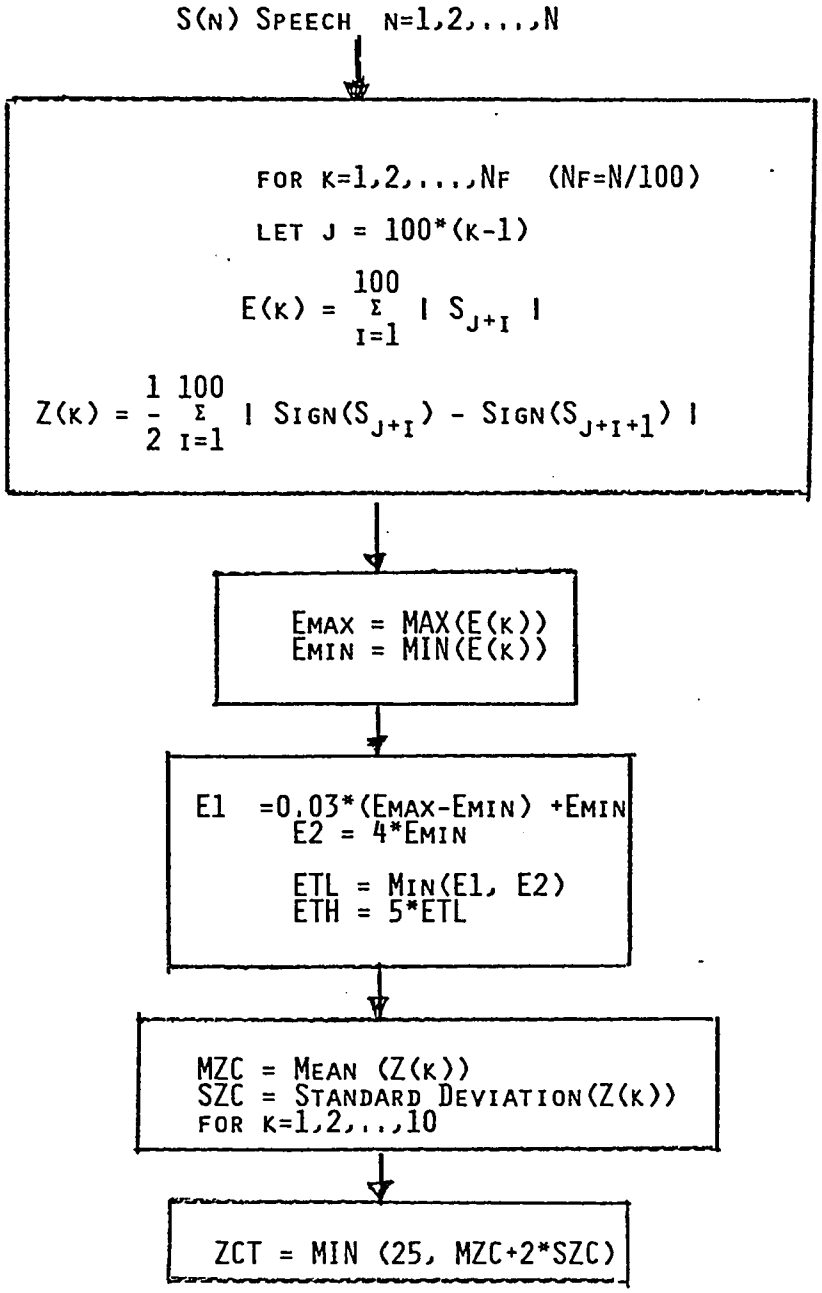
THE MERGING ALGORITHM



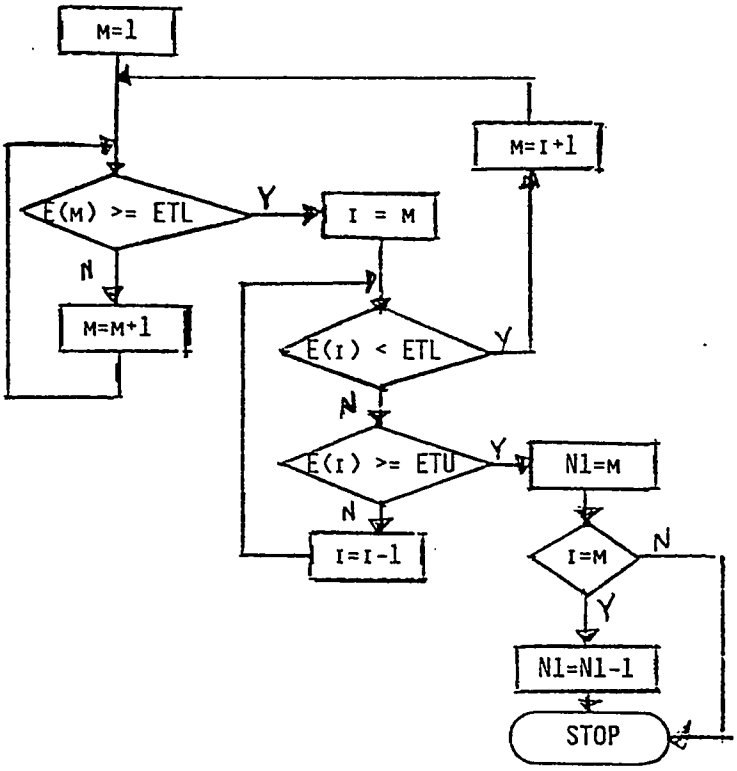
APPENDIX D

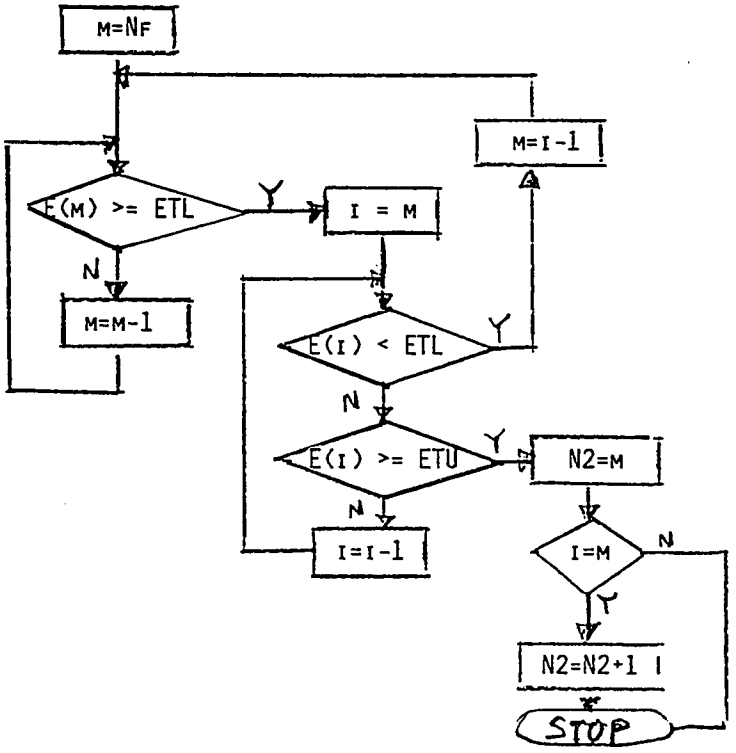
The End Point Detection Algorithm

(A) The determination of Energy and Zero-crossing rate thresholds.



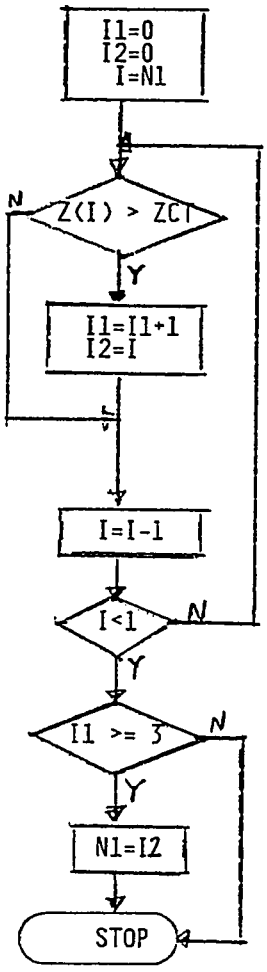
(B) DETERMINATION OF LOWER AND UPPER ENDS (N1,N2) FROM ENERGY.



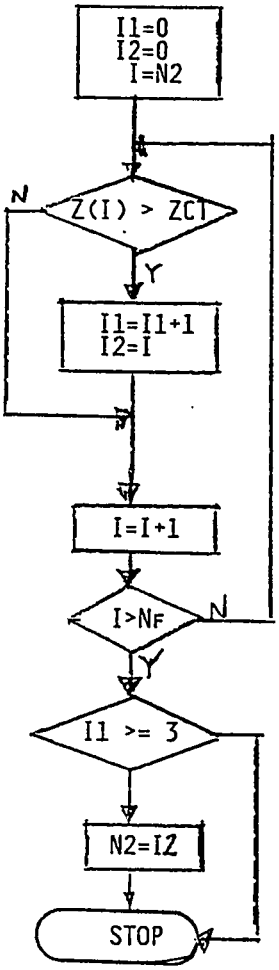


(C). UPDATE N1 AND N2 USING ZERO_CROSSING RATES.

(1). LOWER END

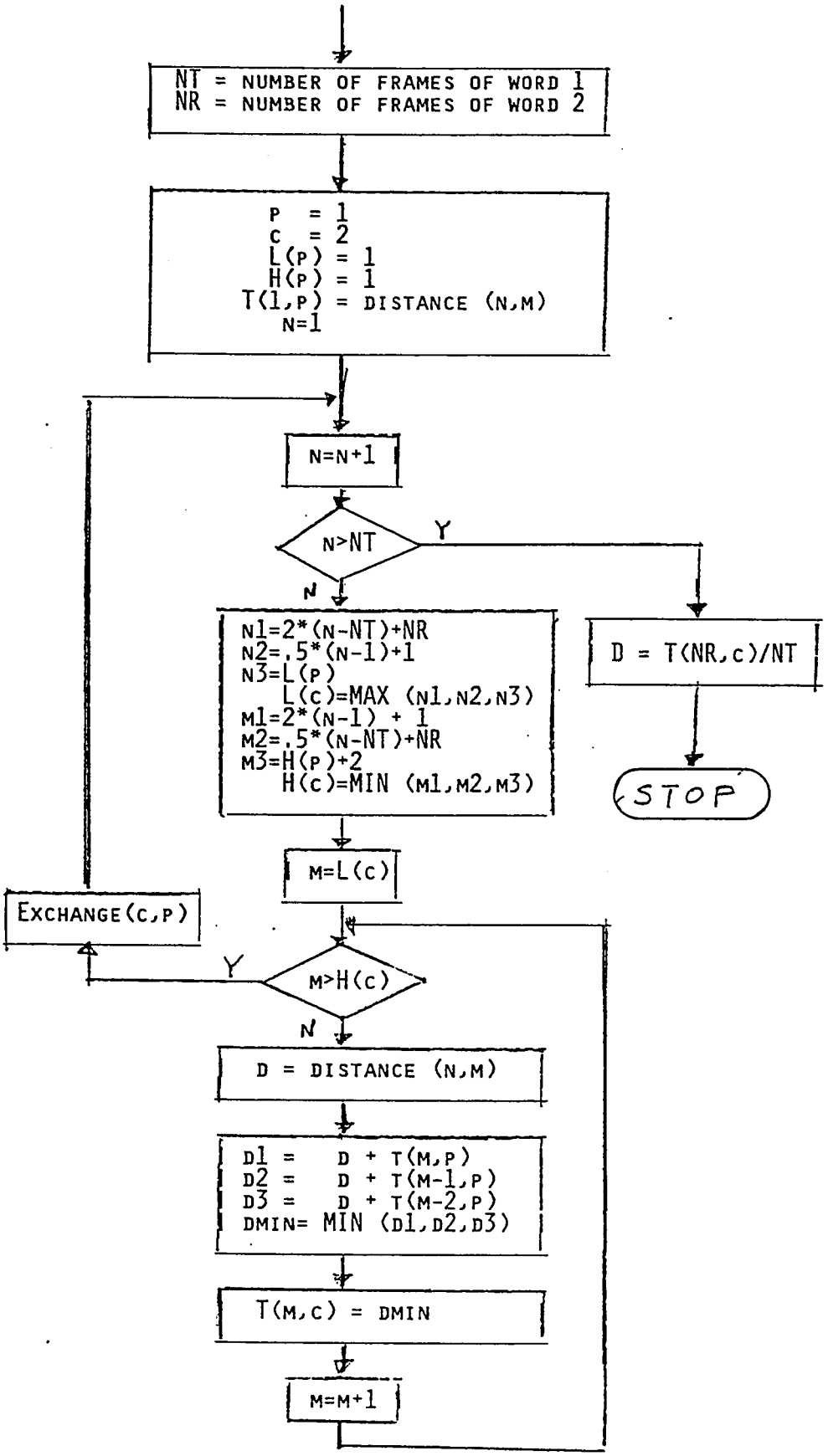


(2). UPPER END



APPENDIX E

THE DYNAMIC TIME-WARPING ALGORITHM


NT = NUMBER OF FRAMES OF WORD 1
NR = NUMBER OF FRAMES OF WORD 2
P = 1
C = 2
L(P) = 1
H(P) = 1
T(1,P) = DISTANCE (N,M)
N=1
N=N+1
N>NT
Y
N
N1=2*(N-NT)+NR
N2=.5*(N-1)+1
N3=L(P)
L(C)=MAX (N1,N2,N3)
M1=2*(N-1) + 1
M2=.5*(N-NT)+NR
M3=H(P)+2
H(C)=MIN (M1,M2,M3)
D = T(NR,C)/NT
STOP
M=L(C)
EXCHANGE(C,P)
Y
M>H(C)
N
D = DISTANCE (N,M)
D1 = D + T(M,P)
D2 = D + T(M-1,P)
D3 = D + T(M-2,P)
DMIN= MIN (D1,D2,D3)
T(M,C) = DMIN
M=M+1

APPENDIX F.

SPECIFICATION OF THE EQUIPMENTS

SPECIFICATION OF THE EQUIPMENTS

MICROPHONE: SONY F-99T

Type:	One point stereo dynamic microphone
Directivity:	Uni-directional for each of R and L capsules
Frequency response:	80-12000Hz.
Output impedannce:	Low (200ohms)
Output level:	-61dBm

TAPE RECORDER: SONY CFS-88S

Recording System:	4-track 2-channel stereo.
frequency response:	60-13000Hz with Metal Cassettes.
Signal / Noise :	40dB
Total Harmonic distortion:	3%
Wow and Flutter:	40.08% (WRMS)

FILTER: General Radio Company 1952 Universal Filter.

A/D CONVERTER: Tecmar Lab-Master (installed on the IBM PC/XT)

Resolution:	12 bits/sample
Input range:	-10 to +10 volts.
Sampling frequency:	Program controllable maximum of 30Khz

APPPENDIX G

PROGRAM LISTINGS FOR THE FFS IMPLEMENTATION

```
C=================================================
C|                                                =
C|   ISOLATED  WORD RECOGNITION SYSTEM            =
C|                                                =
C|      FIXED FRAME SIZE                          =
C|      KNN DECISION RULE                         =
C|                                                =
C|                                                =
C|   NP = FILTER ORDER                            =
C|   NW = FRAME SIZE(VARIALBE)                    =
C|                                                =
C=================================================
C
C
      COMMON   NP,WINDOW(0:255)
C
      WRITE(6,5)
   05 FORMAT('1ISOLATED WORD RECOGNITION SYSTEM (FFS)', /
     +       '+_____________________________________')
C
      NP=12
      NW=255
      C =(2*3.141592)/NW
      DO 10 K=0,NW
         WINDOW (K)=0.54 - 0.46*COS(C*K)
   10 CONTINUE
C
      OPEN (UNIT=08,ERR=20,ACCESS='DIRECT',RECL=3332,
     &      STATUS='OLD')
      OPEN (UNIT=09,ERR=20,ACCESS='DIRECT',RECL=3588,
     &      STATUS='OLD')
      OPEN (UNIT=10,ERR=20,ACCESS='DIRECT',RECL= 404,
     &      STATUS='OLD')
      GO TO 30
C
   20 WRITE(6,25)
   25 FORMAT(' ABEND DUE TO FILE OPEN ERROR')
      STOP
C
   30 READ(5,40) MODE
   40 FORMAT(I2)
C
      IF (MODE.EQ.1) THEN
         CALL TRAIN
      ELSE IF (MODE.EQ.2) THEN
         CALL CLUSTR
      ELSE IF (MODE.EQ.3) THEN
         CALL TEST
      ELSE
         WRITE(6,50),MODE
   50    FORMAT(' ABEND DUE TO INVALID MODE ',I2)
      ENDIF
C
      STOP
      END
```

```
C=================================================
C|                                                =
C| TRANINING STAGE:                               =
C|    IN THIS PHASE AN UTTERANCE IS PROCESSED AND =
C|    RELEVANT FEATURES, THE DATA AUTOCORRELATION, =
C|    LPC AUTOCORRELATION AND LPC RESIDUALS FOR ALL =
C|    FRAMES,  ARE EXTRACTED AND STORED IN A FILE. =
C|                                                =
C=================================================
C
      SUBROUTINE TRAIN
C
      COMMON   NP,WINDOW(0:255)
      DIMENSION  R(64,0:12), A(64,0:12), RESDUE(64)
      DIMENSION  NWORDS(0:9)
C
      WRITE(6,10)
 10   FORMAT('-  TRAINING MODE', /
     +       '   _____________')
C
      DO 20 I=0,9
 20   NWORDS(I)=0
      NTOTAL = 0
C
 30   CALL PROCES (R,A,RESDUE,NF,NTALKR,
     +             NWORD,NUTTER,NEOF)
      IF (NEOF.EQ.0) THEN
         NWORDS(NWORD)=NWORDS(NWORD)+1
         NTOTAL = NTOTAL + 1
         KEY = NWORD*80 + (NTALKR-1)*4 + NUTTER
         WRITE (8,REC=KEY)  NF,( (R(N,K), K=0,NP ),
     +                           N=1,NF)
         WRITE (9,REC=KEY)  NF,( (A(N,K), K=0,NP ),
     +                        RESDUE(N), N=1,NF)
         GO TO 30
      ENDIF
C
      WRITE(6,40) (I, NWORDS(I),I=0,9 )
 40   FORMAT('-  WORD       NUMBER OF UTTERANCES', /
     +       '   ____       ____________________',
     +       10(/,4X,I1,14X,I2) )
C
      WRITE(6,50) NTOTAL
 50   FORMAT('-  TOTAL WORDS ADDED  = ', I5)
C
      RETURN
      END
```

```
C=====================================================
C|                                                    =
C|    TESTING PHASE:                                  =
C|       THE UNKNOWN INPUT UTTERANCE IS READ, THE     =
C|       RELEVANT FEATURES EXTRACTED AND THEN THE     =
C|       DISTANCE FROM ALL STORED TEMPLATES OF A      =
C|       GIVEN WORD CALCULATED USING DYNAMIC TIME     =
C|       WARPING AND THE AVERAGE OF THE 3 SMALLEST    =
C|       DISTANCES IS STORED AS THE DISTANCE OF THE   =
C|       UTTERANCE FROM THE GIVEN WORD.  THE INPUT    =
C|       UTTERANCE IS CLASSIFIED AS COMING FROM THE   =
C|       WORD THAT RESULTED IN THE LEAST DISTANCE     =
C|       IF THAT DISTANCE IS LESS THAN THE REJECTION  =
C|       THRESHOLD FOR THE GIVEN WORD.                =
C|                                                    =
C=====================================================
C
      SUBROUTINE TEST
C
      COMMON   NP,WINDOW(0:255)
      DIMENSION  A1(64,0:12),R1(64,0:12),RES1(64)
      DIMENSION  A2(64,0:12),R2(64,0:12),RES2(64)
      DIMENSION  D(100), REJECT(0:9)
      INTEGER    WORD,BASE,LO,HI,CENTER(100),
     +           NWORDS(0:9),NRECOG(0:9)
C
      DATA REJECT / .89202720, .89902782, .94698453,
     +              .82621747, .84125143, .80927855,
     +              .75799131, .74959373, .86931318,
     +              .89476645 /
C
      WRITE(6,10)
 10   FORMAT('-  TESTING MODE',/
     +       '+  ____________')
C
      DO 20 I=0,9
         NRECOG(I)=0
         NWORDS(I)=0
 20   CONTINUE
      NUMTST = 0
      NUMREC = 0
      NUMREJ = 0
      NUMMIS = 0
C
      KNN=3
 40   CALL PROCES (R1, A1, RES1, N1, NTALKR,
     +             NWORD,NUTTER,NEOF)
      IF (NEOF.GT.0) GO TO 80
      NWORDS(NWORD)=NWORDS(NWORD)+1
      NUMTST=NUMTST+1
      DMAX  = 1.E50
      DMIN  = DMAX
```

```
      DO 70  WORD = 0,9
        READ (10,REC=WORD+1)NCL,(CENTER(K),K=1,NCL)
        BASE=80*WORD
        DO 50  I = 1,NCL
          KEY = BASE+CENTER (I)
          READ (8,REC=KEY)N2,((R2(N,K),K=0,NP),N=1,N2)
          READ (9,REC=KEY)N2,((A2(N,K),K=0,NP ),
     +                              RES2(N),N=1,N2)
          CALL DTW(R1,A1,RES1,N1,R2,A2,RES2,N2,DIST)
          D(I) = DIST
   50   CONTINUE
        CALL SORT (D,NCL)
        SUM=D(1)
        NUM=1
        DO 60 I=2,KNN
          IF (D(I).LT.100.0) THEN
             SUM=SUM+D(I)
             NUM=NUM+1
          ENDIF
   60   CONTINUE
        DIST=SUM/NUM
        IF (DIST.LT.DMIN) THEN
           DMIN=DIST
           IWORD=WORD
        ENDIF
   70 CONTINUE
C
C--CHECK IF THE LEAST DISTANCE IS SMALLER THAN
C     THE REJECTION THRESHOLD FOR THE WORD.
C
      IF (DMIN .LE.  REJECT(IWORD) ) THEN
        IF (NWORD.EQ.IWORD) THEN
           NRECOG(NWORD)=NRECOG(NWORD)+1
           NUMREC = NUMREC+1
        ELSE
           NUMMIS = NUMMIS+1
           WRITE(6,73) NWORD, NTALKR, IWORD
   73      FORMAT(' ',' WORD =',I4,' BY SPEAKER = ',
     +              I2,' RECOGNIZED AS = ',I2 )
        ENDIF
      ELSE
        NUMREJ = NUMREJ+1
        WRITE(6,74) NWORD, NTALKR,IWORD
   74   FORMAT(' WORD =',I2,' BY = ',I2,' NEAREST TO',
     +         I2,' REJECTED')
      ENDIF
      GO TO 40
```

```
   80  CONTINUE
       WRITE(6,90) (I, NWORDS(I), NRECOG(I),I=0,9),
      +             NUMTST, NUMMIS, NUMREJ, NUMREC
   90  FORMAT('-  WORD     TESTED    RECOGNIZED ',/
      +       ' ____     ______    __________ ',
      +       10(/,4X,I1,9X,I3,10X,I3),                /
      +       '-  TOTAL WORDS TESTED     = ', I4, /
      +       '   TOTAL WORDS MISSED     = ', I4, /
      +       '   TOTAL WORDS REJECTED   = ', I4, /
      +       '   TOTAL WORDS RECOGNIZED = ', I4  )
C
       RETURN
       END



       SUBROUTINE SORT (D,N)
C
       DIMENSION  D(1)
C
       DO 50 I=1,N
          DM=D(I)
          MP=I
          DO 40 J=I,N
             IF (D(J) .LT. DM) THEN
                DM=D(J)
                MP=J
             ENDIF
   40     CONTINUE
          T    = D(I)
          D(I) = DM
          D(MP)= T
   50  CONTINUE
C
       RETURN
       END
```

```
C==== CLUSTERING ALGORITHM
C
      SUBROUTINE CLUSTR
C
      COMMON   NP
      DIMENSION  R1(64,0:12),A1(64,0:12),RES1(64)
      DIMENSION  R2(64,0:12),A2(64,0:12),RES2(64)
      DIMENSION  D(80,80)
      INTEGER    OLD,NEW,TYPE(100),ID(100),W(0:100,2)
      INTEGER    CENTER(100), BASE, COUNT(64)
C
      WRITE(6,05)
  05  FORMAT('-  CLUSTERING  MODE',/
     +       '+  ________________')
C
      DO 10 I = 1,64
  10  COUNT(I)=0
C
  20  READ (5,*,END=999)   IWORD, NTOKEN
      BASE = 80*IWORD
C
C--CALCULATE DISTANCE MATRIX BETWEEN THE UTTERANCES
      DO 90 I = 1,NTOKEN
         KEY1 = BASE + I
         READ (8,REC=KEY1) N1,(( R1(N,K),K=0,NP),
     +                          N=1,N1)
         READ (9,REC=KEY1) N1,(( A1(N,K),K=0,NP),
     +                          RES1(N), N=1,N1)
         D(I,I) = 0.0
         DO 80 J = I+1,NTOKEN
           KEY2 = BASE + J
           READ (8,REC=KEY2) N2,((R2(N,K),K=0,NP),
     +                          N=1,N2)
           READ (9,REC=KEY2) N2,((A2(N,K),K=0,NP),
     +                         RES2(N), N=1,N2)
           IF (N1.GT.N2) THEN
            CALL DTW(R1,A1,RES1,N1,R2,A2,RES2,N2,DIST)
           ELSE
            CALL DTW(R2,A2,RES2,N2,R1,A1,RES1,N1,DIST)
           ENDIF
           D(I,J) = DIST
           D(J,I) = DIST
  80     CONTINUE
  90  CONTINUE
```

```
C--CLUSTER THE TOKENS
C
        THRESH =0.6
        KMAX   = 5
        DO 100 I=1,NTOKEN
          TYPE (I)=0
 100    CONTINUE
C
        NOUTL = 0
        NCL   = 0
        NOTCL = NTOKEN
 110    IF (NOTCL.EQ.0) GO TO 910

C--ISOLATE UNCLUSTERED TOKENS
        J = 0
        DO 150 I=1,NTOKEN
          IF (TYPE (I) .EQ. 0) THEN
             J = J+1
             ID(J) = I
          ENDIF
 150   CONTINUE

C--FIND MINIMAX CENTER  OF THE UNCLUSTERED TOKENS
        DMIN = 1.E30
        DO 300 I=1,NOTCL
            I1 = ID (I)
            DMAX = 0
            DO 200 J=1,NOTCL
               J1 = ID (J)
               DMAX = MAX (D(I1,J1), DMAX)
 200        CONTINUE
            IF (DMAX .LT. DMIN) THEN
               DMIN = DMAX
               MINMAX = I1
            ENDIF
 300    CONTINUE
C
        NEW = 2
        OLD = 1
C
C--INITIALIZE OLD CLUSTER SET TO EMPTY SET
        W(0,OLD)=NOTCL
        DO 400 I=1,NOTCL
           W(I,OLD)=0
 400    CONTINUE
```

```
        DO 800 K=1,KMAX

C--FIND CLUSTER SET FOR THE MINIMAX CENTER
          N = 0
          DO 500 I=1,NOTCL
             I1 = ID (I)
             IF ( D(MINMAX, I1 ).LT.THRESH) THEN
                N = N+1
                W (N,NEW) = I1
             ENDIF
 500      CONTINUE
          W(0,NEW) = N

C--FIND MINIMAX CENTER  FOR THE NEW CLUSTER SET
          DMIN = 1.E30
          DO 700 I = 1,N
             DMAX = 0
             I1  = W(I,NEW)
             DO 600 J=1,N
                J1  = W(J,NEW)
                DMAX = MAX ( D (I1,J1), DMAX)
 600         CONTINUE
             IF (DMAX .LT. DMIN) THEN
                DMIN = DMAX
                MINMAX = I1
             ENDIF
 700      CONTINUE
C
C--COMPARE NEW AND OLD CLUSTER SETS.  EXIT IF SAME
          DO 750 I = 0,N
             IF (W(I,OLD).NE. W(I,NEW) ) GO TO 760
 750      CONTINUE
          GO TO 850
C
 760      I   = OLD
          OLD = NEW
          NEW = I
 800    CONTINUE
C
C--SAVE THE NEW CLUSTER CENTER IF NOT AN OUTLIER
 850    IF (N. GT. 1) THEN
           NCL = NCL + 1
           CENTER (NCL) = MINMAX
           COUNT(NCL)=N
        ELSE
           NOUTL = NOUTL + 1
        ENDIF
C
C--ELIMINATE MEMBERS OF THE NEW CLUSTER
        DO 900 I = 1,N
           J  = W (I,NEW)
           TYPE (J) = 1
 900    CONTINUE
        NOTCL = NOTCL-N
        GO TO 110
```

```
 910    WRITE (10,REC=IWORD+1)NCL,(CENTER(K),K=1,NCL)
C
      WRITE(6,920)  IWORD, NTOKEN, NOUTL, NCL
 920  FORMAT('-  WORD CLUSTERED     = ',I3, /
     +       '0  CANDIDATE TOKENS   = ',I3, /
     +       '0  NUMBER OF OUTLIERS = ',I3, /
     +       '0  NUMBER OF CLUSTERS = ',I3  )
C
      WRITE(6,930)  (I,COUNT(I),I=1,NCL )
 930  FORMAT('-  CLUSTER CENTER     TOKENS IN CLUSTER',
     +       20 (/,' ', 8X,I2,18X,I2) )
        GO TO 20
C
 999  RETURN
      END
```

```
C=================================================
C|                                                =
C|                                                =
C|  DYNAMIC TIME-WARPING ALGORITHM                 =
C|                                                =
C|  INPUTS:                                        =
C|     R1   DATA AUTOCORRELATION OF TEST UTTERANCE =
C|     A1   LPC  AUTOCORRELATION OF TEST UTTERANCE =
C|     RES1 LPC RESIDUALS OF THE TEST UTTERANCE    =
C|     NT   NUMBER OF FRAMES OF THE TEST UTTERANCE.=
C|                                                =
C|     R2   DATA AUTOCORRELATION OF REFERENCE      =
C|     A2   LPC  AUTOCORRELATION OF REFERENCE      =
C|     RES2 LPC RESIDUALS OF THE REFERENCE         =
C|     NR   NUMBER OF FRAMES OF THE REFERENCE      =
C|                                                =
C|  OUTPUT:                                        =
C|                                                =
C|     DIST: THE DISTANCE BETWEEN THE TWO UTTERANCES=
C|           FOR THE OPTIMAL DYNAMIC TIME WARPED   =
C|           PATH ASSIGNMENT.                      =
C|                                                =
C=================================================
C
      SUBROUTINE DTW (R1,A1,RES1,NT,R2,A2,RES2,NR,DIST)
C
      COMMON   NP
      DIMENSION  R1(64,0:12),A1(64,0:12),RES1(64)
      DIMENSION  R2(64,0:12),A2(64,0:12),RES2(64)
      DIMENSION  TOTAL(64,2)
      INTEGER    SWITCH(64,2), LO(2),HI(2),OLD,NEW
C
      OLD = 1
      NEW = 2
      DO 100 I=1,NR
          TOTAL  (I,OLD) = 0.0
          TOTAL  (I,NEW) = 0.0
          SWITCH (I,OLD) = 0
          SWITCH (I,NEW) = 0
  100 CONTINUE
C
      IDEL   = 2
      LO(OLD)= 1
      HI(OLD)= 1+IDEL
C
      DO 150 I=LO(OLD),HI(OLD)
        TOTAL(1,I)=DISTFR(R1,A1,RES1,R2,A2,RES2,1,I)
  150 CONTINUE
```

```
      DO 500 N = 2,NT
C
C--OBTAIN THE POSSIBLE FRAMES OF THE REFERENCE
C  TO BE MATCHED TO THE N'TH FRAME OF THE TEST
          IF (N .LE. (1+IDEL) ) THEN
            LO(NEW) = MAX ( 1, LO(OLD) )
          ELSE
            M1 = 2*(N-NT)+(NR-IDEL)
            M2 = .5*(N-(1+IDEL)+1) + 1
            M3 = LO(OLD)
            LO(NEW) = MAX (M1, M2, M3)
          ENDIF
          IF (N .GE. (NT-IDEL) ) THEN
            HI(NEW) = MIN ( NR, HI(OLD)+2 )
          ELSE
            M1 = 2*(N-1)+(1+IDEL)
            M2 =.5*(N-(NT-IDEL))+NR
            M3 = HI(OLD)+2
            HI(NEW) = MIN ( M1, M2, M3)
          ENDIF
          DO 200 J = 1,HI(NEW)
             SWITCH(J,NEW) = 0
             TOTAL (J,NEW) = 0.0
 200      CONTINUE
          DIST = 1.E30
          DO 400 M=LO(NEW),HI(NEW)
            D1 = DISTFR (R1,A1,RES1,R2,A2,RES2,N,M)
            DMIN = 1.E30
            DO 300 K=M-2,M
              IF (K.LT.LO(OLD) .OR. K.GT.HI(OLD) )
     +                    GO TO 300
              IF (SWITCH(K,OLD).EQ. 1 .AND. K.EQ.M)
     +                    GO TO 300
              S1 = D1 + TOTAL(K,OLD)
              IF (S1 .LT. DMIN) THEN
                 DMIN = S1
                 J    = K
              ENDIF
 300        CONTINUE
            TOTAL (M,NEW) = DMIN
            IF (J .EQ. M)  SWITCH(M,NEW) = 1
            IF (DMIN .LT. DIST) THEN
              DIST = DMIN
              NODE = M
            ENDIF
 400      CONTINUE
          IF (NODE .EQ. NR) GO TO 600
          I   = OLD
          OLD = NEW
          NEW = I
 500  CONTINUE
      N=NT
 600  DIST = DIST/N
C
      RETURN
      END
```

```
C=======================================================
C|                                                      =
C| THE FUNCTION DISTFR CALCULATES THE DISTANCE BETWEEN  =
C| TWO FRAMES OT THE TWO UTTERANCES USNING THE ITAKURA  =
C| LOG LIKELIHOOD RATIO.                                =
C|                                                      =
C|  INPUTS:                                             =
C|     R1   DATA AUTOCORRELATION OF UTTERANCE ONE       =
C|     A1   LPC  AUTOCORRELATION OF UTTERANCE ONE       =
C|     RES1 LPC RESIDUALS OF UTTERANCE ONE              =
C|     N    FRAME OF UTTERANCE ONE.                     =
C|                                                      =
C|     R2   DATA AUTOCORRELATION OF UTTERANCE TWO       =
C|     A2   LPC  AUTOCORRELATION OF UTTERANCE TWO       =
C|     RES2 LPC RESIDUALS OF UTTERANCE TWO              =
C|     M    FRAME OF UTTERANCE TWO.                     =
C|                                                      =
C|                                                      =
C|  OUTPUT:                                             =
C|                                                      =
C|     DISTFR: THE DISTANCE BETWEEN THE TWO FRAMES.     =
C|                                                      =
C=======================================================
C
      FUNCTION DISTFR (R1, A1, RES1, R2, A2, RES2, N, M)
C
      COMMON   NP
C
      DIMENSION  A2(64,0:12),R2(64,0:12),RES2(64)
      DIMENSION  A1(64,0:12),R1(64,0:12),RES1(64)
C
      E1 = 0
      E2 = 0
      DO 10 J=1,NP
         E1 = E1 + R1(N,J) * A2(M,J)
         E2 = E2 + R2(M,J) * A1(N,J)
 10   CONTINUE
      E1 = 2*E1 + R1(N,0) * A2(M,0)
      E2 = 2*E2 + R2(M,0) * A1(N,0)
      D1 = E1/RES1(N)
      D2 = E2/RES2(M)
      D1 = ALOG(D1)
      D2 = ALOG(D2)
      DISTFR = .5*(D1+D2)
C
      RETURN
      END
```

```
C=================================================
C|                                                =
C| THIS SUBROUTINE PERFORMS THE FOLLOWING FUNCTIONS =
C|                                                =
C|    1.  READS A TOKEN (SAMPLE)                  =
C|    2.  CONVERTS THE SIGNAL VALUES TO THEIR     =
C|        CORRECT NUMERIC VALUES                  =
C|    3.  CALCULATES THE ENDPOINTS                =
C|    4.  PREMPAHSISES THE DATA                   =
C|    5.  BLOCKS THE DATA INTO FIXED LENGTH FRAMES. =
C|    6.  APPLIES A HAMMING WINDOW TO THE FRAMES  =
C|    7.  CALCULATES THE AUTOCORRELATION OF THE   =
C|        WINDOWED FRAME                          =
C|    8.  CALLS THE LPC SUBROUTINE TO PERFORM THE =
C|        LPC ANALYSIS.                          =
C|                                                =
C=================================================
C
      SUBROUTINE PROCES (R, A, RESDUE, NF, NTALKR,
     +                   NWORD, NUTTER, NEOF)
C
      COMMON   NP,WINDOW(0:255)
      DIMENSION  R(64,0:12),A(64,0:12), RESDUE(64)
      DIMENSION  X(0:511),S(10000), E(110), IZ(110)
      INTEGER*2  HIB(10000),LOB(10000)
      INTEGER    FSIZE, WSIZE, HI
C
      NSAMPL = 8192
C
C--------READ A SAMPLE WORD
      READ(7,110,END=120)(HIB(N),LOB(N),N=1,NSAMPL+2)
 110  FORMAT(255(Z1,Z2) )
      GO TO 130
 120  NEOF=1
      RETURN
C
 130  NWORD  = HIB(NSAMPL+1)
      NUTTER = LOB(NSAMPL+1)
      NTALKR = LOB(NSAMPL+2)
      NEOF   = 0
C
C---------CONVERT THE VALUES
      C = 1./204.8
      DO 140 N=1,NSAMPL
          HI = HIB(N)
          LO = LOB(N)
          IF (HI.GE.8) HI = HI+240
          IVALUE = 256*HI+LO
          IF (IVALUE.GT.32767) IVALUE=IVALUE-65536
          S(N) = C*IVALUE
 140  CONTINUE
C
C FIND ENDPOINTS
C
C======  CALCULATE BOTH THE ENERGY AND ZERO-CROSSING
C        RATES FOR A WINDOW OF 12.5 MS (100 SAMPLES)
      NF  = 0
```

```
      EMAX= 0.
      EMIN= 1.E45
C
      WSIZE=100
      DO 20 I=1,NSAMPL-WSIZE,WSIZE
         SUM = 0
         IZZ = 0
         NI  = I+WSIZE-1
         DO 10 J=I,NI
           SUM = SUM + ABS (S(J))
           IF ((S(J).LE.0 .AND. S(J+1).GT.0) .OR.
     +      (S(J).GT.0 .AND. S(J+1).LE.0) ) IZZ=IZZ+1
   10    CONTINUE
         NF = NF +1
         IZ(NF)= IZZ
         E(NF) = SUM
         EMAX  = MAX (E(NF), EMAX)
         EMIN  = MIN (E(NF), EMIN)
   20 CONTINUE
      EMIN  = MAX (EMIN, .03)
C
      E1 = 0.03*(EMAX-EMIN)+EMIN
      E2 = 4*EMIN
      EL = MIN (E1, E2)
      EU = 5*EL
C
      S1=0
      S2=0
      DO 30 J=1,10
         S1 = S1 + IZ (J)
         S2 = S2 + IZ (J) * IZ (J)
   30 CONTINUE
      IZMEAN = S1/10
      IZSDEV = SQRT( (S2 -10*IZMEAN*IZMEAN)/9 )
      IZCT   = MIN (25, IZMEAN+2*IZSDEV )
      IZCT   = MAX (10, IZCT)
C
C--------CALCULATE THE LOWER END FROM ENERGY
      M = 1
   40 IF (E(M).LT.EL) THEN
         M=M+1
         GO TO 40
      ENDIF
      I = M
   50 IF (E(I).GE.EL .AND. E(I).LT.EU) THEN
         I=I+1
         GO TO 50
      ENDIF
      IF (E(I).LT.EL) THEN
         M=I+1
         GO TO 40
      ENDIF
      LO = M
      IF (I.EQ.M) LO=LO-1
C
C--------CALCULATE THE UPPER END FROM ENERGY
      M = NF
```

```
   60  IF (E(M).LT.EL) THEN
          M=M-1
          GO TO 60
       ENDIF
       I = M
   70  IF (E(I).GE.EL .AND. E(I).LT.EU) THEN
          I=I-1
          GO TO 70
       ENDIF
       IF (E(I).LT.EL) THEN
          M=I-1
          GO TO 60
       ENDIF
       HI = M
       IF (I.EQ.M) HI=HI+1
C
C--------UPDATE THE LOWER END BY ZERO-CROSSING -
       I1 = 0
       I2 = 0
       I3 = LO-LO/2
       DO 80 I=LO,I3,-1
          IF (IZ(I).GE.IZCT) THEN
             I1 = I1+1
             I2 = I
          ENDIF
  80   CONTINUE
       IF (I1.GE.3)  LO=I2
C
C--------UPDATE THE UPPER END BY ZERO-CROSSING -
       I1 = 0
       I2 = 0
       I3 = HI+(NF-HI+1)/2
       DO 90 I=HI,I3
          IF (IZ(I).GE.IZCT) THEN
             I1 = I1+1
             I2 = I
          ENDIF
  90   CONTINUE
       IF (I1.GE.3) HI=I2
C
  100  LO = 100*(LO-1)+1
       HI = 100*HI    -1
       LO = MAX (LO, 1)
       HI = MIN (HI, NSAMPL)
C
C--------PREMPHASIZE DATA
       T1=S(LO)
       DO 200 K=LO+1,HI
          T2  = S(K)
          S(K)= S(K)-.95*T1
          T1  = T2
  200  CONTINUE
C
C
       FSIZE = 128
       WSIZE = 255
       NF    = (HI-LO+1)/FSIZE
```

```
C
      DO 700 N=1,NF
C--APPLY A HAMMING WINDOW TO A FRAME
         LB = LO+(N-1)*FSIZE
         DO 400 K=0,WSIZE
            X(K) = S(LB+K) * WINDOW(K)
 400     CONTINUE
C
C--CALCULATE THE DATA AUTOCORRELATION OF A SINGLE FRAME
         DO 600 K=0,NP
            SUM = 0.
            NK  = WSIZE-K
            DO 500 NY=0,NK
               SUM = SUM + X(NY)*X(NY+K)
 500        CONTINUE
            R(N,K)=SUM
 600     CONTINUE
 700  CONTINUE
      CALL LPC (R, A, RESDUE, NF)
C
      RETURN
      END
C
C=============================================================
C|                                                           =
C|    IMPLEMENTATION OF THE DURBIN ALGORITHM TO CALCULATE    =
C|    THE LPC PARAMETERS FOR ALL FRAMES OF THE UTTERANCE     =
C|                                                           =
C|    INPUT:                                                 =
C|        R     THE DATA AUTOCORRELATION MATRIX              =
C|        NF    THE NUMBER OF FRAMES OF THE UTTERANCE        =
C|                                                           =
C|    OUTPUT:                                                =
C|         A    THE LPC AUTOCORRELATION MATRIX               =
C|         RESIDUE  THE VECTOR OF LPC RESIDUALS FOR ALL      =
C|                  FRAMES OF THE UTTERANCE.                 =
C|                                                           =
C=============================================================
C
      SUBROUTINE LPC (R, A, RESDUE, NF)
C
      COMMON      NP
      DIMENSION   R(64, 0:12), A(64,0:12), RESDUE(64)
      DIMENSION   RC(0:21)
C
      DO 400 N=1,NF
C
C--NORMALIZE THE DATA AUTOCORRELATION
         TEMP  = R(N,0)
         R(N,0)= 1.0
         DO 10 K=1,NP
            R(N,K)=R(N,K)/TEMP
  10     CONTINUE

C--APPLY DURBIN'S RECURSIVE ALGORITHM TO OBTAIN LPC'S
         A(N,0) = 1.0
         A(N,1) = -R(N,1)/R(N,0)
```

```
      RC(1)  = A(N,1)
      ALPHA  = R(N,0) * ( 1.0 - A(N,1)*A(N,1) )
      DO 50 I=2,NP
        I1 = I-1
        Q  = R(N,I)
        DO 30 K=1,I1
           Q = Q + A(N,K)*R(N,I-K)
30      CONTINUE
        Q = -Q/ALPHA
        RC(I) = Q
        I2 = I/2
        DO 40 K=1,I2
           J  = I-K
           T1 = A(N,K) + Q*A(N,J)
           T2 = A(N,J) + Q*A(N,K)
           A(N,K) = T1
           A(N,J) = T2
40      CONTINUE
        A(N,I) = Q
        ALPHA  = ALPHA*(1.0 - Q*Q)
        IF (ALPHA.LE.0.0) THEN
           WRITE(6,45)
45         FORMAT(' PROGRAM TERMINATED DUE TO INSTABILITY')
           STOP
        ENDIF
50    CONTINUE
      RESDUE(N) = ALPHA
C
C--FIND AUTOCORRELATION OF LPC PARAMETERS
      DO 200 I=0,NP
        NPU = NP-I
        Q = 0.0
        DO 150 K=0,NPU
           Q = Q + A(N,K)*A(N,K+I)
150     CONTINUE
        RC(I) = Q
200   CONTINUE
C
      DO 300 K=0,NP
         A(N,K)=RC(K)
300   CONTINUE
400 CONTINUE
C
      RETURN
      END
```

APPENDIX H

SAMPLE OUTPUTS FOR THE FFS IMPLEMENTATION

ISOLATED WORD RECOGNITION SYSTEM(FIXED FRAME SIZE)

TESTING MODE

SPEAKERS NOT USED IN TRAINING PHASE

WORD = 8 BY SPEAKER = 23 RECOGNIZED AS = 0
WORD = 0 BY SPEAKER = 23 RECOGNIZED AS = 9
WORD = 6 BY SPEAKER = 24 RECOGNIZED AS = 9
WORD = 7 BY SPEAKER = 24 RECOGNIZED AS = 4

WORD	NUMBER TESTED	NUMBER RECOGNIZED
0	5	4
1	5	5
2	5	5
3	5	5
4	5	5
5	5	5
6	5	4
7	5	4
8	5	4
9	5	5

TOTAL WORDS TESTED = 50
TOTAL WORDS MISSED = 4
TOTAL WORDS REJECTED = 0
TOTAL WORDS RECOGNIZED = 46

ISOLATED WORD RECOGNITION SYSTEM

TESTING MODE

TESTING WORDS USED IN TRAINING

WORD = 1 SPEAKER = 3 REJECTED NEAREST TO 9
WORD = 1 BY SPEAKER = 5 RECOGNIZED AS = 7

WORD	NUMBER TESTED	NUMBER RECOGNIZED
0	0	0
1	34	32
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0

TOTAL WORDS TESTED = 34
TOTAL WORDS MISSED = 1
TOTAL WORDS REJECTED = 1
TOTAL WORDS RECOGNIZED = 32

ISOLATED WORD RECOGNITION SYSTEM (FIXED FRAME SIZE)

TESTING MODE

SPEAKERS USED IN TRAINING PHASE
UTTERANCES NOT USED IN TRAINING
WORD = 4 BY SPEAKER = 7 RECOGNIZED AS = 7

WORD	NUMBER TESTED	NUMBER RECOGNIZED
0	2	2
1	2	2
2	2	2
3	2	2
4	2	1
5	2	2
6	2	2
7	2	2
8	2	2
9	2	2

TOTAL WORDS TESTED = 20
TOTAL WORDS MISSED = 1
TOTAL WORDS REJECTED = 0
TOTAL WORDS RECOGNIZED = 19

BIBLIOGRAPHY

1. J. L. Flanagan, "Computers that Talk and Listen: Man-Machine Communication by Voice," Proc. IEEE, vol. 64, pp. 405-415, Apr. 1976.
2. W. Lea, "The value of Speech Recognition Systems," in [8], pp. 3-18.
3. T. B. Martin, "Practical Applications of Voice Input to Machines," Proc. IEEE, vol. 64, pp. 487-501, Apr. 1976.
4. L. R. Rabiner and R. W. Schafer, "Digital Techniques for Computer Voice Response: Implementation and Applications," Proc. IEEE, vol. 64, pp. 416-433, Apr. 1976.
5. A. E. Rosenberg, "Automatic Speaker Verification: A Review," Proc. IEEE, vol. 64, pp. 460-475, Apr. 1976.
6. B. S. Atal, "Automatic Recognition of Speakers From Their Voices," Proc. IEEE, vol. 64, pp. 460-475, Apr. 1976.
7. Jean-Paurl Haton Ed., Automatic Speech Analysis and Recognition Dordrecht, Holland: D. Reidel Publishing Company, 1982.
8. W. Lea, Ed., Trends In Speech Recognition. Englewood Cliffs, NJ: Prentice Hall, 1980.
9. D. R. Reddy, Ed., Speech Recognition. New York: Academic, 1974.

10. N. R. Dixon and T. B. Martin, Eds., Automatic Speech and Speaker Recognition. New York: IEEE Press, 1979.

11. L. R. Rabiner and R. W. Schafer, Digital Processing of Speech Signals. Englewood Cliffs, NJ: Prentice Hall, 1978.

12. W. Lea, "Speech Recognition: Past, Present and Future," in [8], pp. 39-98.

13. D. R. Reddy "Speech Recognition by Machine: A Review," Proc. IEEE, vol. 64, pp. 501-531, Apr. 1976.

14. J. T. Tou and R. C. Gonzalez, Pattern Recogniton Principles. Reading, MA: Addison-Wesley, 1974.

15. L. R. Rabiner and M. R. Sambur, "An Algorithm for Determining the Endpoints of Isolated utterances," The Bell system Technical Journal, vol. 54, no.2, pp. 297-315, Feb. 1975.

16. L. F. Lamel, L. R. Rabiner, A. E. Rosenberg and J.G. Wilpon, "An Improved Endpoint Detector for Isolated Word Recognition," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-29, pp. 777-785, Aug. 1981.

17. B. S. Atal and S. L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," Journal of the Acoustical Society of America, vol. 50, no.2, pp. 637-655, 1971.

18. J. D. Markel and A. H. Gray Jr., Linear Prediction of Speech. New York: Springer verlag, 1976.

19. J Makhoul, "Linear Prediction: A Tutorial Review," Proc. IEEE, vol. 63, pp. 561-580, Apr. 1975.

20. F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-23, pp. 67-72, Feb. 1975.

21. A. H. Gray, Jr. and J. D. Markel, "Distance Measures for Speech Processing," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-24, pp. 380-391, Oct. 1976.

22. H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word recognition," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-26, pp. 43-49, Feb. 1978.

23. L. R. Rabiner, A. E. Rosenberg and S. E. Levinson, "Considerations in Dynamic Time-warping Algorithms for Discrete Word Recognition," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-26, pp. 575-582, Dec 1978.

24. C. S. Myers, L. R. Rabiner and A. E. Rosenberg, "Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-28, pp. 622-635, Oct. 1980.

25. L. R. Rabiner, "On Creating Reference Templates for Speaker Independent Recognition of Isolated Words," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-26, pp. 34-42, Feb. 1978.

26. S. E. Levinson, L. R. Rabiner, A. E. Rosenberg and J.G. Wilpon, "Interactive Clustering Techniques for

Selecting Speaker Independent Reference Templates for Isolated Word Recognition," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-27, pp. 134-141, Apr. 1979.

27. L. R. Rabiner and J.G.xWilpon, "Considerations in Applying Clustering Techniques to Speaker Independent Word Recognition," Journal of Acoustical Society of America, vol. 66(3), pp. 663-673, Sept. 1979.

28. M. R. Sambur and L. R. Rabiner, "A Speaker Independent Digit-Recognition System," The Bell system Technical Journal, vol. 54, no.1, pp. 81-102, Jan. 1975.

29. L. R. Rabiner, S. E. Levinson, A. E. Rosenberg and J.G.xWilpon, "Speaker-Independent Recognition of Isolated Words Using Clustering Techniques," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-27, pp. 336-349, Aug. 1979.

30. G. M. White and R. B. Neely, "Speech Recogntion Experiment With Linear Prediction, Bandpass Filtering and Dynamic Programming," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-24, pp. 183-188, Apr. 1976.

31. V. N. Gupta, J. K. Bryan and J. N. Gowdy, "A Speaker Independent Speech Recognition System based on Linear Prediction," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-26, pp. 27-33, feb. 1978.

32. L. R. Rabiner and S. E. Levinson, "Isolated and Connected Word Recognition--Theory and Selected Applications," IEEE Transactions on Communications, vol. COM. 29, no.5, pp. 621-659, May. 1981.

33. J. J. Wolf "Speech Recognition and Understanding," in Digital Pattern Recogntion, Edited by K. S. Fu, Heidelberg, Germany: Springer Verlag, 1976.

34. Ronald A. Cole, et. al., "Feature-Based Speaker Independent Recognition of Isolated English Letters," ICASSP 83, vol. 2, pp 731-733.

35. A. J. Newel, J. Barnet, J. W. Forgie, C.C. Green, D.H. Klatt, J. Munson, D.R. Reddy, W.A. Woods, Speech Understanding Systems: Final Report of a Study Group. North Holland/American Elsevier, 1973.

36. J. J. Wolf and W. A. Woods, "The HWIM Speech Understanding System," in [8], pp. 316-339.

37. J. Barnet, M. I. Bernstein, R. Gillman and Iris Kameny, "The SDC Speech Understanding System," in [8], pp. 272-293.

38. B. Lowerre and Raj Reddy, "The Harpy Speech Understanding System," in [8], pp. 340-360.

39. L. D. Erman and V. R. Lesser, "The Hearsay-II Speech Understanding System: A Tutorial," in [8], pp. 361-381.

40. D. H. Klatt "Overview of the ARPA Speech Understanding Project," in [8], pp. 249-271.

41. F. Jelinek, "Continuous Speech Recognition by Statistical Methods," Proc. IEEE, vol. 64, pp. 532-556, Apr. 1976.

42. L.R. Bahl, J.K. Baker, P.S. Cohen, A.G. Cole, F. Jelinek, B.L. Lewis and R.L. Mercer "Automatic Recognition of Continuously Spoken Sentences From a Finite State Grammar," ICASSP 78, pp 418-421.

43. F. Jelinek, "Self-Organized Continuous Speech Recognition," in [7], pp. 231-238.

44. W. A. Lea and June E. Shoup, "Specific Contributions of the ARPA SUR Project," in [8], pp. 382-421.

45. Raj Reddy and Victor Zue, "Recognizing Continuous Speech Remains an Elusive Goal," IEEE Spectrum, pp. 84-87, Nov. 1983.

46. N. Levinson "The Weiner RMS (root mean square) Error Criterion in Filter Design and Prediction," Appendix B, in N. Weiner, Extrapolation, Interpolation and Smoothing of Stationary Time Series, Cambridge, MS: MIT Press, 1949.

47. J. Durbin "The Fitting of Time-series Models," Review of Institute of Internationa Statistics, vol. 28, no. 3, pp. 223-243, 1960.

48. B. S. Atal, "Linear Prediction of Speech--Recent Advances With Applications to Speech Analysis," in [9], pp. 221-230.

49. V. R. Viswanathan, J. Makhoul, R. M. Schwartz and A.W.F. Huggins, "Variable Frame rate Transmission: A Review of Methodology and Application to Narrow-Band LPC Speech Coding," IEEE Transactions on Communications, vol. COM. 30, no.4, pp. 674-686, April 1982.

50. R. Pieraccini and R. Billi, "Experimental Comparison Among Data Compression Techniques in Isolated Word Recognition," ICASSP 83, vol. 3, pp 1035-1028.

51. J. L. Gauvain, J. Mariani and J. S. Lienard, "On the Use of Time Compression for Word-Based Recognition," ICASSP 83, vol. 3, pp 1029-1032.

52. C. K. Chuang and S. W. Chan, "Speech Recognition Using Variable Frame Rate Coding," ICASSP 83, vol. 3, pp 1033-1036.

53. F. J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," Proc. IEEE, vol. 66, pp. 51-82, Jan. 1978.

54. B. A. Dautrich, L. R. Rabiner, T. B. Martin, "On the Use of Filter Bank Features for Isolated Word Recogntion," ICASSP 83, vol. 3, pp 1061-1064.

55. L. R. Rabiner and J.G.xWilpon, "Isolated Word Recognition Using a Two-Pass Pattern Recognition Approach," ICASSP 81, vol. 2, pp 724-731.